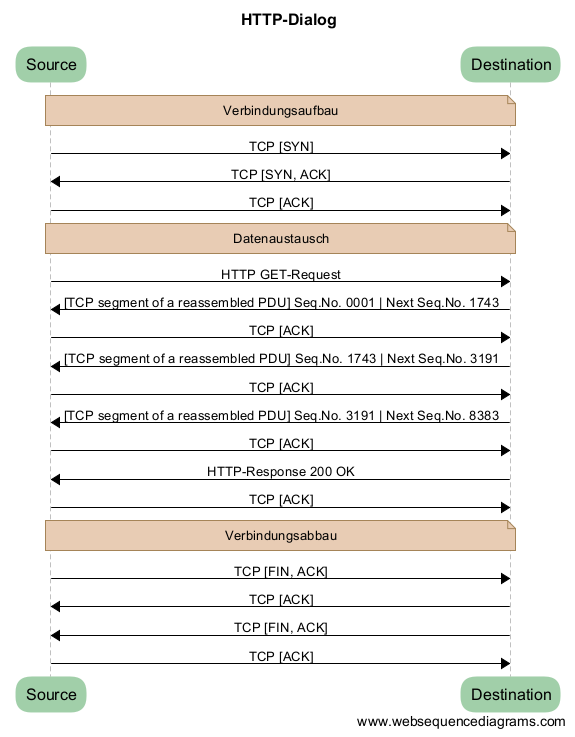
Protokollteil 1:



Der http-Dialog lässt sich grob in drei Phasen aufteilen: Verbindungsaufbau, Datenaustausch und Verbindungsabbau. Der Verbindungsaufbau erfolgt mittels des Drei-Wege-Handschlags in TCP: Der Client sendet dem Server ein SYN-Paket, was der Server mit einem SYN/ACK-Paket bestätigt und dem Verbindungsaufbau zustimmt. Der Client bestätigt das mit einem ACK-Paket.

Am Anfang des Datenaustausches sendet der Client ein http-Request an den Server. Dieser sendet nun die Antwort in mehreren Segmenten, welche einzeln vom Client bestätigt werden. Im letzten Segment wird ein http-Response mit dem Code 200 vom Server an den Client gesendet, wo die einzelnen Segmente zusammengesetzt werden. Das wird ebenfalls vom Client mit einem ACK-Paket bestätigt.

Beim Verbindungsabbau sendet der Client ein FIN/ACK-Paket an den Server, um die Verbindung Clientseitig zu beenden. Dieser bestätigt dies mit einem ACK-Paket und sendet ebenfalls ein FIN/ACK-Paket und beendet so die Verbindung Serverseitig, was zum Schluss vom Client mit einem ACK-Paket bestätigt wird.

Betrachtet man nun das http-Request Paket genauer, werden hier die verschiedenen Schichten klar:

|  |  |
| --- | --- |
|  | TCP/IP-Modell |
| Frame | Network Interface |
| Ethernet II | Network Interface |
| Internet Protocol Version 4 | Internet |
| Transmission Control Protocol | Transport |
| Hypertext Transfer Protocol | Application |

Das Frame ist die physikalische Darstellung des gesamten Pakets. Im Ethernet werden die Quell- und Zieladresse als MAC-Adresse dargestellt. Da diese nicht direkt in einer IPv4 Adresse gespeichert werden können, erfolgt eine Zuweisung über das Address Resolution Protocol (ARP). Über TCP werden die Art und Weise der Kommunikation zwischen Quell- und Zieladresse festgelegt. Mittels http werden letztendlich die Daten ausgetauscht.

Wie kann Wireshark aber nun die einzelnen Segmente des http-Responses zusammensetzen? Das wird durch einen genaueren Blick auf das TCP Layer der jeweiligen Pakete deutlich. Jedes Paket beinhaltet eine sogenannte „Sequence Number“. Die „[TCP segment of a reassambled PDU]“ Pakete beinhalten darüber hinaus auch ein Feld für „Next Sequence Number“. Über diese wird die relative Reihenfolge der Pakete genau angegeben.

Protokollteil 2:

Generell sind neben den IP Adressen und Ports auch der Verbindungsaufbau ersichtlich. Wo dann allerdings beim vorher dargestellten http-Dialog der Nachrichtenaustausch in Form eines http-Request anfängt, versenden wir über TLSv1.2 im Secure Sockets Layer ein „Handshake Protocol: Client Hello“ und erhalten ein ACK-Paket, gefolgt von „Handshake Protocol: Server Hello“ vom Server. Als nächstes überträgt der Server uns in mehreren Segmenten ein Zertifikat, welches der Client mit einem ACK-Paket bestätigt. Der Client sendet drauf hin „Client Key Exchange, Change Cipher Spec“. Die weitere Kommunikation ist ab hier verschlüsselt.

|  |  |  |
| --- | --- | --- |
| Protokolle | TCP/IP-Modell | OSI-Referenzmodell |
| Frame | Network Interface | Layer 1: Physical |
| Ethernet II | Network Interface | Layer 2: Data Link |
| Internet Protocol Version 4 | Internet | Layer 3: Network |
| Transmission Control Protocol | Transport | Layer 4: Transport |
| Secure Sockets Layer | Transport | Layer |

Die Verschlüsselung betrifft nur die überliegenden Schichten, also 5-7 im OSI-Modell und das Application Layer im TCP/IP-Modell. Die darunterliegenden Schichten bleiben dabei lesbar.

Protokollteil 3:

Beim Aufruf der http-Adresse werden wir auf die <https://www.haw-hamburg.de/ti-i.html> weitergeleitet. Rufen wir direkt die https-Adresse auf, werden wir aufgefordert das Zertifikat zu akzeptieren und gelangen dann zu einem Raumbuchungssystem der HAW. Da wir beim http-Aufruf Port 80 anfragen, gibt uns der Server via http den Code „302 Found“ zurück mit einem Verweis auf die oben genannte Adresse, welche dann direkt aufgerufen wird. Mit dem https-Aufruf wird allerdings direkt der Port 443 adressiert, wo es Serverseitig keine Weiterleitung auf die allgemeine Website gibt.

Protokollteil 4:

Mit dem Befehl „ifconfig“ konnten wir neben dem Interface auch die IP-Adresse und Netzmaske abrufen. Die Netzadresse und Broadcastadresse haben wir manuell berechnet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Interface | IP-Adresse | Netzmaske | Netzadresse | Broadcastadresse |
| eth0 | 141.22.27.104 | 255.255.254.0 | 141.22.26.0 | 141.22.27.255 |
| eth1 | 192.168.18.131 | 255.255.255.0 | 192.168.18.0 | 192.168.18.255 |
| eth2 | 172.16.1.7 | 255.255.255.0 | 172.16.1.0 | 172.16.1.255 |
| lo | 127.0.0.1 | 255.0.0.0 | 127.0.0.0 | 127.255.255.255 |

Beim ping unserer Loopback-Adresse haben wir selbst – wie erwartet - auch wieder geantwortet.

Bei einen ping auf die Broadcast-Adresse 141.22.27.255 haben verschiedene Teilnehmer des Subnetzes geantwortet, wobei häufig ein „(DUP!)“ für Duplicate als Hinweis mit ausgegeben wurde.

Zuerst gab es bei einen ping auf die Broadcast-Adresse 192.168.18.255 keine Antwort, da die entsprechenden Netzwerkkomponenten nicht eingeschaltet waren. Nachdem das von der Praktikumsleitung behoben wurde, gab es entgegen unserer Erwartung nur zwei Teilnehmer die antworteten. Einsicht bat uns der Netzwerk Plan von Herrn Hartmut Schulz. Zum einen stellten wir fest, dass nur ein ISDN-Router und ein Switch antworteten. Zum anderen das alle anderen lokalen Teilnehmer im selben Netz ausgeschaltet waren.

Ein ping auf die Broadcast-Adresse 172.16.1.255 ergab erneut keinerlei Antworten anderer Teilnehmer, obwohl wir mithilfe eines Mitstudierenden, der Praktikumsleitung und Wireshark bestätigen konnten, dass die ping Anfragen bei anderen Teilnehmern ankamen. Im weiteren Verlauf des Praktikums konnte leider nicht geklärt werden, weshalb es keine Antwort auf den ping gab.