



Laborübung - HTTP 1.x

04.06.2018

In dieser Laborübung sollen Sie die beiden HTTP-Versionen 1.0 und 1.1 mit und ohne Pipelining analysieren. Hierzu kommen Ihr über Docker realisierter Webserver und die index.htm mit 4 eingebetteten Bilder zum Einsatz. Als Ersatz dienen die im Labor bereitgestellten Webserver-Container.

Aufgabe 1 - HTTP mit CURL

Curl ist ein nützliches Tool, um auf einfache Art und Weise Dateien über verschiedene Protokolle zu übertragen, wie z.B. über HTTP. So wird Curl gerne beim Testen von HTTP-Seiten und -Anwendungen genutzt.

Booten Sie Lubuntu und öffnen Sie ein Terminal.

- a) Beginnen Sie mit der Website aus der vorangegangenen TCP-Übung. Rufen Sie über Curl die Webseite ab, in dem Sie folgenden Befehl in das Terminal eingeben:
`curl http://www2.inf.h-bonn-rhein-sieg.de`

Wie viele eingebettete Objekte beinhaltet die Webseite?

- b) Wenden Sie Anfrage auf Ihren eigenen Webserver bzw. auf 10.30.0.108 an.
- c) Lassen Sie sich über Curl den Header der HTTP-Antwort ausgeben. Welche HTTP-Version wurde bei der Kommunikation verwendet?

Aufgabe 2 - Analyse der HTTP 1.0-Kommunikation

Eine HTTP-Kommunikation über die Version HTTP 1.0 einzusetzen, ist leider nicht mehr so einfach. Zur Erklärung:

Über welche HTTP-Version Browser und Server kommunizieren, wird bei der Kommunikation über den HTTP-Header ausgehandelt. Der Browser schlägt bei der HTTP-Anfrage eine von ihm unterstützte Version vor. In der Antwort schreibt der Server dann die für die Kommunikation gültige Version in den Header. In der Regel ist das die, die der Browser vorgeschlagen hat.

Die meisten Browser und Webserver unterstützen schon kein HTTP 1.0 mehr. Der Firefox (Mozilla) und der Apache bilden hier die Ausnahme. Jedoch selbst wenn Firefox die HTTP-Version 1.0 anfordert, setzt der Apache die Version automatisch auf HTTP 1.1 hoch, denn er erkennt anhand der Browserversion, dass der Firefox durchaus HTTP 1.1 unterstützt. Deshalb muss der Apache über einen Befehl angewiesen werden, dass er bei einer Mozilla -Anfrage strikt die Version HTTP 1.0 verwendet.

- a) Fügen Sie in Ihr Dockerfile den Befehl `RUN echo 'BrowserMatch "Mozilla" downgrade-1.0 force-response-1.0' >> /etc/apache2/apache2.conf` ein, bauen Sie Ihr Image neu und starten Sie Ihren Container.
- b) Starten Sie nun den Firefox und geben Sie in die Adresszeile `about:config` ein. Hier können Sie den Firefox weitergehend konfigurieren als über den Menüpunkt „Einstellungen“. Geben Sie in die Suchzeile `http.version` ein und



ändern Sie die HTTP-Version von 1.1 zu 1.0. Starten Sie den Browser neu und starten Sie anschließend Wireshark. Deaktivieren Sie alle reassembling-Optionen zum HTTP-Protokoll in den Wireshark-Einstellungen. Starten Sie die Capture-Funktion und rufen Sie die Webseite zu Ihrem Webserver bzw. zu 10.30.0.107 auf. Danach schließen Sie den Browser und stoppen das Capture.

- c) Wie viele Verbindungen werden zum Abruf der Webseite benötigt?
- d) Wie viele Pakete werden insgesamt versendet?
- e) Wie viel Zeit benötigt der Aufruf der Webseite?
Hinweis: Zur Messung eignen sich besonders die Firefox oder Chrom Developer Tools

Aufgabe 3 - Analyse der HTTP 1.1-Kommunikation ohne Pipelining

- a) Entfernen Sie den unter Aufgabe 2 eingesetzten Browser-Match wieder aus Ihrem Dockerfile, bauen Sie Ihr Image neu und starten Sie Ihren Container.
- b) Starten Sie den Firefox und stellen Sie wieder die Version 1.1 über die `about:config` ein. Starten Sie die Capture-Funktion des Wiresharks und rufen Sie die Webseite zu Ihrem Container bzw. zu 10.30.0.108 auf. Danach schließen Sie den Browser und stoppen das Capture.
- c) Wie viele Verbindungen werden zum Abruf der Webseite benötigt?
- d) Wie viele Pakete werden insgesamt versendet?
- e) Wie viel Zeit benötigt der Aufruf der Webseite?
- f) Welche Unterschiede können Sie zu dem in der Theorie dargestellten Kommunikationsablauf ausmachen?

Viel Spaß und Erfolg!