

Praktikum 2: Web-Plattform, Infrastruktur

Gerrit Burkert, 06.09.2015

Ziele

In diesem Praktikum werden Sie HTTP untersuchen und Werkzeuge zum Dateitransfer und zum Arbeiten mit einer *Remote Shell* kennenlernen.

- Installation einiger Werkzeuge
- Per Telnet “von Hand” mit dem Webserver kommunizieren
- HTTP mit Hilfe einer Browser-Erweiterung genauer untersuchen
- Webseite per SFTP auf den Laborserver *dublin* laden
- Mit SSH an einem Server anmelden, dort arbeiten und Dateien editieren

Installationen

Als ersten soll sichergestellt werden, dass alle benötigten Werkzeuge verfügbar sind.

Browser und Entwicklerwerkzeuge

Dieser Teil wurde bereits im letzten Praktikum erledigt. In diesem Praktikum werden wir die HTTPFox-Erweiterung des Firefox benutzen.

Telnet

Telnet steht unter Windows, auf dem Mac und unter Linux normalerweise zur Verfügung. Unter Windows muss es möglicherweise erst aktiviert werden. Mehr dazu in der Aufgabe unten.

Dateitransfer

Um Daten zwischen Client und Server zu kopieren benötigen Sie ein Werkzeug für den Dateitransfer. Es sollte mindestens die beiden Protokolle FTP und SFTP beherrschen.

Nicht sehr übersichtlich aufgebaut und kaum schön anzusehen ist das grafische Oberfläche von **FileZilla**. Trotzdem handelt es sich bei FileZilla um ein robustes FTP- und SFTP-Programm, weshalb es hier empfohlen werden soll. Eine Alternative ist **Cyberduck** (Win, Mac).

- <https://filezilla-project.org>
- <http://cyberduck.ch>

Neben Programmen mit grafischer Oberfläche ist es von Vorteil, auch Kommandozeilenwerkzeuge für den Dateitransfer zur Verfügung zu haben. Auf Unix-Systemen (Linux, Mac) können Sie normalerweise einfach `ftp` bzw. `sftp` in einem Terminal aufrufen. Unter Windows können Sie zu diesem Zweck die entsprechenden Putty-Programme verwenden.

- <http://www.putty.org>

Remote Shell

Schliesslich sollten Sie sich auch auf einem Server einloggen können, um dort Dateien zu verschieben, zu löschen, oder direkt zu bearbeiten. Dazu verwenden wir `ssh`, um eine verschlüsselte Verbindung aufzubauen. Auf Unix-Systemen (Linux, Mac) können Sie normalerweise einfach `ssh`. Unter Windows installieren Sie zu diesem Zweck am besten Putty.

- <http://www.putty.org>

Aufgabe 1: HTTP

Der Browser kommuniziert mit dem Webserver über das **Hypertext Transfer Protocol (HTTP)**. Dieses Protokoll soll nun etwas genauer angesehen werden.

Telnet

In einem ersten Versuch werden wir per HTTP direkt im **Telnet** mit dem Webserver kommunizieren. Dazu öffnen wir auf Port 80 eine Verbindung zum Server. Dann können wir Text – in diesem Fall einen HTTP-GET-Request – zum Server schicken. Die Antwort des Servers wird dann im Telnet-Programm angezeigt.

Auf dem Mac und unter Linux sollten Sie Telnet einfach durch Eingeben des Befehls `telnet` in einem Terminal starten können. Unter Windows müssen Sie Telnet erst aktivieren (Beschreibung für Windows 7): *Start → Programme und Funktionen → Windows Funktionen aktivieren/deaktivieren*. In der Liste die Funktion Telnet-Client (nicht Telnet-Server!) suchen, aktivieren und mit OK bestätigen. Falls das nicht klappt, können Sie auch *putty* verwenden, indem Sie statt dem *open*-Befehl (s.u.) den *dublin.zhaw.ch* als Host und Port 80 angeben, das Protokoll auf *raw* setzen und bei Close Window *never* einstellen.

- Starten Sie `telnet`
- Stellen Sie eine Verbindung zum Server her:
`open dublin.zhaw.ch 80`
- Senden Sie das HTTP-Kommando an den Server (anschliessend zweimal die Eingabetaste drücken)
`GET /~bkrt/hallo.html HTTP/1.0`
- Nun sollte der Server die HTML-Datei schicken
- Probieren Sie es auch mit der Datei `index.html` der Beispiel-Website
- Machen Sie einen Screenshot des Terminalfensters

Hinweise:

- Beim Eingeben des HTTP-Befehls sollten Sie sich nicht vertippen. Ein Korrigieren mit der Backspace-Taste funktioniert normalerweise nicht.
- Nach dem `open`-Befehl im Windows-Terminal kann der Cursor auf eine Zeile springen, die bereits mit Text belegt ist. Entweder Sie überschreiben den Text einfach oder Sie drücken vor der Eingabe von `GET . . .` ein paarmal die Eingabetaste, bis der Cursor auf einer freien Zeile ist.

Hier ist ein typischer Ablauf des HTTP-Requests:

```
burkert — bash — 80x30
bash-3.2$ telnet
telnet> open dublin.zhaw.ch 80
Trying 160.85.67.138...
Connected to srv-app-t-001.zhaw.ch.
Escape character is '^]'.
GET /~bkrt/hallo.html HTTP/1.0

HTTP/1.1 200 OK
Date: Sat, 28 Sep 2013 17:09:40 GMT
Server: Apache/2.2.15 (CentOS)
Last-Modified: Thu, 26 Sep 2013 13:24:42 GMT
ETag: "5b0040-af-4e7494ae51a80"
Accept-Ranges: bytes
Content-Length: 175
Connection: close
Content-Type: text/html; charset=UTF-8

<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Hallo</title>
  </head>
  <body>
    <h1>Hallo</h1>
    <p>Ich bin eine Webseite</p>
  </body>
</html>
Connection closed by foreign host.
bash-3.2$
```

HTTP via Telnet

HttpFox

Etwas komfortabler können Sie die HTTP-Header in der Firefox-Erweiterung HttpFox ansehen:

- Öffnen Sie HttpFox im Firefox-Browser
- Starten Sie die Protokollierung durch Klicken auf den *Start*-Button
- Öffnen Sie die gewünschte Seite auf dem *dublin* im Browser
- Klicken Sie auf den *Stop*-Button

Sie sollten nun die HTTP-Requests zum Server in der Liste sehen. Wenn Sie *hallo.html* angefordert haben, wurde (abgesehen vom Favicon) nur die HTML-Datei angefordert. Probieren Sie auch einen Zugriff auf <http://dublin.zhaw.ch/~bkrt/web1/index.html>. Hier sollten Sie auch noch Zugriffe auf *screen.css* und *logo.jpg* sehen.

- Wählen die Zeile mit dem Zugriff auf die HTML-Datei aus

- Untersuchen Sie den Request- und den Response-Header
- Machen Sie einen Screenshot der beiden Header
- Was könnte der Grund sein, dass im Request-Header der Host auftaucht, obwohl die TCP-Verbindung zum Host bereits aufgebaut ist, wenn der HTTP-Request geschickt wird?

curl (fakultativ)

Falls auf Ihrem System das Kommandozeilentool *curl* installiert ist, können Sie auch dieses noch ausprobieren (oder bei Interesse installieren):

- Finden Sie heraus, wozu *curl* gut ist. Tipp: Unter Unix gibt es sogenannte *man pages*, die Informationen zu Unix-Befehlen liefern. Diese sind auch übers Web zugänglich, in diesem Fall hilft es, <http://man.cx/curl> aufzurufen.
- Versuchen Sie, die Datei *hallo.html* mit *curl* herunterzuladen.

Aufgabe 2: Dateitransfer

Die Themen *Dateitransfer* und *Remote Shell* werden wir im Unterricht erst später im Semester behandeln. Sehen Sie die praktische Übung in dieser und der nächsten Aufgabe daher als Vorbereitung auf diesen Theorieteil an.

Für die Übertragung einer Website auf einen Server, wird FTP oder SFTP verwendet:

- **FTP (File Transfer Protocol):** Das File Transfer Protocol [...] ist ein im RFC 959 von 1985 spezifiziertes Netzwerkprotokoll zur Übertragung von Dateien über IP-Netzwerke. [...] 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. (Quelle: Wikipedia)
- **SFTP (SSH File Transfer Protocol):** Das SSH File Transfer Protocol oder Secure File Transfer Protocol (SFTP) ist eine für die Secure Shell (SSH) entworfene Alternative zum File Transfer Protocol (FTP), die Verschlüsselung ermöglicht. (Quelle: Wikipedia)

Da bei FTP auch die Logindaten unverschlüsselt übertragen werden, sollten Sie wenn immer möglich SFTP verwenden. Es gibt zahlreiche S/FTP-Programme.

Website Deployment

Damit eine Website im Internet abrufbar ist, muss sie von einem Webserver als Antwort auf einen HTTP-Request ausgeliefert werden. Einen solchen Webserver stellt der Laborserver `dublin.zhaw.ch` zur Verfügung. Erklärungen zu den Services auf diesem Server finden Sie auf der Startseite <http://dublin.zhaw.ch>.

- Stellen Sie mit Ihrem SFTP-Client eine Verbindung zum Server her. Wenn der Port nicht vorgegeben ist, geben Sie 22 ein, das ist der SSH-Port.
- Bei der ersten Verbindung sollte der SFTP-Client nach einer Bestätigung des Fingerabdrucks für den Server fragen. Können Sie sich vorstellen, warum das nötig ist?
- Wenn die Verbindung zum Server hergestellt ist, wechseln Sie ins Verzeichnis `www`.
- Dort erstellen Sie ein neues Verzeichnis `web1` (FileZilla: Kontextmenü).
- Laden Sie die Beispiel-Website aus dem letzten Praktikum auf den Server. Achten Sie darauf, dass die Verzeichnisstruktur erhalten bleibt und die Datei `index.html` direkt im Verzeichnis `web1` liegt.

Wenn Sie diese Schritte durchgeführt haben, sollte die Website unter folgender Adresse abrufbar sein (*kurzzeichen* ersetzen Sie durch Ihr ZHAW-Kurzzeichen):
`http://dublin.zhaw.ch/~kurzzeichen/web1/index.html`

Machen Sie ein paar weitere Versuche:

- Testen Sie, ob die Seite auch erscheint, wenn Sie in der URL den Dateinamen *index.html* weglassen.
- Mit welcher URL können Sie direkt auf das Bild *logo.jpg* zugreifen?

Kommandozeilentools (fakultativ)

Da Sie in der letzten Teilaufgabe mit grosser Wahrscheinlichkeit einen SFTP-Client mit grafischer Oberfläche (FileZilla, Cyberduck) verwendet haben, wäre es noch interessant, die gleichen Schritte mit Kommandozeilen-Tools durchzuführen.

Finden Sie heraus, wie Sie auf der Kommandozeile Dateien per *sftp* und per *scp* (secure copy) kopieren können und probieren Sie es aus, indem Sie Dateien zwischen ihrem Notebook und dem dublin-Server hin- und herkopieren. Unter Windows verwenden Sie die Putty-Tools *psftp* und *pscp*.

Aufgabe 3: Remote Shell

Sie wissen nun, wie Sie Dateien mit Hilfe eines SFTP-Clients auf den Server laden oder vom Server zu Ihrem lokalen Notebook kopieren. Oft ist es aber auch sinnvoll, Dateien direkt auf dem Server bearbeiten oder andere Aufgaben auf dem Server ausführen zu können (zum Beispiel einen Serverprozess zu starten). Dazu stellen wir per SSH eine verschlüsselte Verbindung zum Server her und öffnen dort ein Kommandozeilenprogramm (Shell).

- Stellen Sie eine ssh-Verbindung zum *dublin*-Server her_
 - Auf dem Mac geben Sie dazu `ssh dublin.zhaw.ch -l kurzzeichen` ein. Wenn Ihr Benutzername auf dem Notebook dem Kurzzeichen entspricht, können Sie die Option `-l` ... auch weglassen.
 - Unter Windows verwenden Sie *putty*, geben den Server und Port 22 an, sowie *SSH* als Verbindungstyp.
- Geben Sie `ls -l` ein, um sich eine Übersicht über die vorhandenen Verzeichnisse zu verschaffen.
- Geben Sie `cd www/web1` ein, um in Ihr zuvor angelegtes Website-Verzeichnis zu wechseln.
- Welche Verzeichnisse und Dateien befinden sich hier?
- Mit `pwd` (print working directory) erhalten Sie den Pfad des aktuellen Verzeichnisses
- Kopieren Sie mit `cp index.html neu.html` die HTML-Datei in eine neue Datei *neu.html*.
- Rufen Sie *neu.html* im Browser auf.
- Öffnen Sie die Datei mit `vi neu.html` zum Bearbeiten.
- Wechseln Sie durch Drücken der Taste `i` in den Einfügemodus.
- Bewegen Sie den Cursor zum `h1`-Element und ändern Sie die Überschrift.
- Wechseln Sie durch Drücken der Taste `ESC` zum Kommandomodus des `vi`.
- Geben Sie das Kommando `:w` zum Sichern der Änderungen ein.
- Geben Sie das Kommando `:q` zum Verlassen des Editors ein.
- Rufen Sie *neu.html* im Browser auf. Ist die Überschrift geändert?
- Löschen Sie mit `rm neu.html` die Kopie der Datei.

Der *vi* (bzw. *vim* für Vi IMproved) hat den Ruf, ein Editor für Profis zu sein. Wie Sie gerade gesehen haben, ist es aber keine Kunst, mit diesem Editor schnell ein paar Änderungen an einer Datei zu machen. Mehr Befehle kann man sich dann nach und nach aneignen. Ein Problem für *vi*-Neueinsteiger ist vor allem die Unterscheidung von Eingabe- und Kommandomodus. Daran hat man sich aber schnell gewöhnt.

Damit Sie sich etwas weiter mit den Unix-Shell-Befehlen und dem *vi* beschäftigen können, finden Sie bei den Praktikumsunterlagen noch zwei Kurzbeschreibungen: *vi_cheat_sheet.pdf* und *unix101.pdf*.