

Walther- Rathenau- Gewerbeschule Freiburg	Ein paar Fragen zu: HTTP Grammatik-Beschreibungssprachen netcat (nc)		Fach: ITS	Gruppe:
			11. Februar 2008	Seite 1
			Name:	
			Klasse: E3IT1	
			Punkte:	Note:

Die RFC2616 enthält eine genaue Definition des Hypertext Transfer Protocols, Verion 1.1. Das Protokoll arbeitet auf Schicht7 (Anwendungsschicht) und basiert wie alle Protokolle der Schicht7 auf einfachen Text-Kommandos.

Aus diesem Grund ist es möglich, einen HTTP-Client (Web-Browser) oder einen HTTP-Server (Webserver) mit dem Unix-Kommando `netcat` (mit `nc` aufrufen) zu simulieren.

Selbstverständlich verrät die man-page zu `nc`, wie man das anstellen muss.

1 Nun die gefürchteten Fragen

Die Antworten bitte notieren, ich sammle alles ein!

1. Die `rfc2616` verwendet die *Augmented BNF* (erweiterte Backus-Naur-Form) um die Grammatikregeln des Protokolls zu definieren.

Auch SQL ist mit der BNF definiert. Z.B.:

```
"SELECT" [ "ALL" | "DISTINCT" ] spaltenausdruck tabellenausdruck
```

Gib alle möglichen, syntaktisch richtigen Ausdrücke des `select`-Befehls an. `'tabellenausdruck'` und `'spaltenausdruck'` sollen dabei stehen bleiben (nicht auflösen!).

2. `rfc2616` erlaubt 2 mögliche Versionen der GET-Anfrage.

Gib beide Möglichkeiten an, mit der die Datei `'index.html'` auf der Maschine `141.31.147.114` an Port `80` angefragt wird.

3. Teste die beiden Varianten mit Hilfe von `netcat` (`nc`). D.h. Ihr sollt mit `nc` die Anfrage eines Webbrowsers an den Webserver auf `141.31.147.114:80` simulieren.

4. Nachdem wir mit `nc` einen Server abgefragt haben, wollen wir uns nun mit `nc` ausgeben lassen, was ein echter Browser als Anfrage an einen Server schickt. Mit welcher Kommandozeile kann man `nc` starten, dass es als Server auf `localhost` läuft und auf Anfragen an Port `80` wartet?

5. Starte `nc` im Server-Modus an Port `9999` und richte eine Browser-Anfrage an `nc`. Bestätige die Anfrage von `nc` aus mit dem nötigen **Antwort-Protokoll** und hänge die untenstehende Datei an. Der Text der Datei lässt sich dabei durch Selektieren mit der Maus und Einfügen (mittlere Maustaste) in `nc` erzeugen. Beende anschliessend `nc` mit `control-c` und beobachte den Web-Browser. Starte `nc` erneut als Server und schicke das Formular nun ab.

Damit Ihr die Datei `hello.html` nicht abschreiben müsst, könnt Ihr sie so von meinem Webserver laden:

```
wget http://msv.wara.de/~dienert/archive/hello.html
```

Damit Ihr nicht jedesmal den Text des Antwort-Protokolls in `nc` eingeben müsst, könnt Ihr die `http`-Antwort natürlich auch vorne an die Datei `hello.html` anfügen.

Beobachte nun was in `nc` geschieht. Wie gelangen die Formulardaten zum Server wenn man das Formular mit der Methode `GET` bzw. `POST` erzeugt (Obiges Prozedere mehrmals mit `GET` und `POST` wiederholen) ? Was passiert mit im Formular eingegebenen Sonderzeichen und Umlauten?

Die Datei hello.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <p>This is a test html document.</p>

    <form action="hello.html" method="post">
      <table>
        <tr>
          <td>
            <input type="text" name="feld1" value="michael"/>
          </td>
          <td>
            <input type="text" name="feld2" value="carla"/>
          </td>
          <td>
            <input type="text" name="feld3" value="johannes"/>
          </td>
        </tr>
        <tr>
          <td>
            <input type="submit"/>
          </td>
        </tr>
      </table>
    </form>

  </body>
</html>
```