Versuche mit HTTP und netcat

Michael Dienert

7. Mai 2015

Inhaltsverzeichnis

1 RFC2616

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.

2 netcat

Laut Handbuchseite ist das Kommando `nc` das *Schweizer Taschenmesser* der Netzwerktechnik.

Mit folgender Kommandozeile kann man sich mit `nc` mit einem tcp-Socket auf einem entfernten Rechner verbinden:

```
nc <ip-adresse oder name des hosts> <gewuenschter tcp-prt>
Beispiel:
nc dt.wara.de 80
```

Das Kommando bewirkt, dass die Standardeingabe (Tastatur) auf den eingestellten Port des entfernten Rechners umgeleitet wird. Die Antwort, die der entfernte Rechner schickt, wird auf die Standardausgabe (Bildschirm) des lokalen Rechners umgeleitet.

Man kann nc auch im Server-Modus starten:

```
nc -l <gewuenschter tcp-port>
Beispiel:
nc -l 8888
```

Mit dem Kommando

```
netstat -an | less
```

Kann man sich anschliessend die tcp/ip-Verbindungen des Rechners anzeigen lassen:

```
nc -l 8888
netstat -an | less
-->
Aktive Internetverbindungen (Server und stehende Verbindungen)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:8888            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:5432          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
...
```

Man erkennt, dass auf dem Rechner die Dienste ssh (22), cups (631), postgres (5432) und smtp (25) laufen. Und eben auch nc, das als Server auf Anfragen an Port 8888 hört.

3 Die beiden Formen eines http-Request

Mit der in de RFC2616 beschriebenen Syntax, ergeben sich 2 gültige Formen eines http-Request:

Die erste Form besteht aus Request- und einer Headerzeile:

```
GET /index.html HTTP/1.1 CRLF
Host: www.google.de CRLF
CRLF
```

Die einzeilige Form sieht so aus:

```
GET http://www.google.de/index.html HTTP/1.1 CRLF CRLF
```

4 Selbst ausprobieren

1. Testen Sie die beiden Varianten
2. Starten Sie nun nc im Server-Modus an Port 9999 und richten Sie eine Browser-Anfrage an nc. Bestätigen Sie die Anfrage von nc aus mit dem nötigen **Antwort-Protokoll** und dem folgenden html-Text. Verwenden Sie dazu den Inhalt der Datei response.txt (vgl. unten).

   Gehen Sie dabei wie folgt vor:

   (a) Eine Konsole (=Terminal) öffnen

   (b) Herunterladen der Datei in die Konsole:

   ```
   wget http://dt.wara.de/response.txt
   ```

(c) Ausgabe der Datei in der Konsole:

```
cat response.txt
```

(d) Selektieren des gewünschten Textes mit der Maus (nur Selektieren!)

(e) Einfügen des Textes an der Stelle des Mauszeigers durch Drücken der mittleren Maustaste.

(f) Abschicken der http-Response durch Drücken von **CTRL-d**

3. Erneutes Starten von nc im Server-Modus
4. Formular im Browser ausfüllen und abschicken
5. Beobachten Sie im nc-Terminal, wie die Formulardaten zum Webserver gelangen.
6. Wiederholen Sie das ganze Spiel, diesmal mit der Methode POST im html-Formular. Wie werden die Daten nun übertragen?

Der http-Response mit Datei hello.html

```
HTTP/1.1 200 OK
Content-Type: text/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="GET">
       <table>
          <tr>
             <td>
                <input type="text" name="feld1" value="wert1"/>
             </td>
             <td>
                <input type="text" name="feld2" value="wert2"/>
             </td>
             <td>
                <input type="text" name="feld3" value="wert3"/>
             </td>
          </tr>
          <tr>
             <td>
                <input type="submit"/>
             </td>
          </tr>
       </table>
    </form>
  </body>
</html>
```