Single Line Web Server With nc on FreeBSD

Brendan Sechter

2016-02-09

I wrote a single file web server in my last post about Getting Started With tor Hidden Services on FreeBSD. I figured it might be useful to have a couple of variations of simple web servers in their own post. This post covers single line and single file web servers with **nc** on FreeBSD. Content served from a static file or a script is presented.

UPDATE: A follow up post exists. A Better Scripted Netcat Server

Software Versions

```
$ date
February  7, 2016 at 02:27:30 AM JST
$ uname -vm
FreeBSD 11.0-CURRENT #0 r287598: Thu Sep 10 14:45:48 JST 2015     root@:/usr/obj/usr/src/sys/MI
```

Instructions

First, add the following to **index.html** so we have a file to serve.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Hello World!</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

The netcat Wikipedia page lists a one-line one-shot web server. It has been adapted for FreeBSD below.

```
FILE=index.html; { printf "HTTP/1.0 200 OK\r\nContent-Length: $(wc -c <$FILE)\r\n\r\n"; cat $FI
```

The server body can be put in a loop to serve the file multiple times. The host and port definitions have been moved to the beginning of the line in the version below.

```
FILE=index.html; HOST=127.0.0.1; PORT=8080; while true; do { printf "HTTP/1.0 200 OK\r\nContent
```

The above line is complicated enough that it might want live in a file, like **server.sh**. A content type definition has been moved to the beginning of the file. A body and response have been defined. A couple of useful headers have been added. Finally, the response number is echoed before each response.

```
#/bin/sh
```

```
HOST=127.0.0.1
PORT=8080
FILE=index.html
CONTENT_TYPE=text/html

while true; do
BODY=$(cat $FILE)
RESPONSE=$(cat <<EOF
HTTP/1.0 200 OK
Content-Type: ${CONTENT_TYPE}
Content-Length: $((${#BODY}+1))
Connection: close

${BODY}
EOF
)
echo "---$((X=X+1))---"
echo "$RESPONSE" | nc -N -l $HOST $PORT
done
```

Make the server script executable before running it.

```
chmod +x server.sh
./server.sh
```

By redefining the body clause, the script can serve dynamic content. A simple JSON date server is listed below.

```
#/bin/sh

HOST=127.0.0.1
PORT=8080
CONTENT_TYPE=application/json

while true; do
BODY=$(cat <<EOF
{
  "Date": "$(date)"
}
EOF
)
RESPONSE=$(cat <<EOF
HTTP/1.0 200 OK
Content-Type: ${CONTENT_TYPE}
Content-Length: $((${#BODY}+1))
Connection: close

${BODY}
EOF
)
echo "---$((X=X+1))---"
echo "$RESPONSE" | nc -N -l $HOST $PORT
done
```

A one line version of the JSON date server is listed below.

```
HOST=127.0.0.1; PORT=8080; CONTENT_TYPE=application/json; while true; do BODY=$(printf "{\r\n
```

A one line version of the single file web server is listed below.

```
HOST=127.0.0.1; PORT=8080; FILE=index.html; CONTENT_TYPE=text/html; while true; do BODY=$(cat $
```

Either of the following lines can be used to test the server from the command line. **date** is piped into **nc** so the client will send EOF to the server. The server will also log something to the terminal this way.

```
curl -v 127.0.0.1:8080
# log request date on the server and send EOF
date | nc 127.0.0.1 8080
```

Note that these servers are not necessarily robust. They may trip up clients that do not close the connection.

References:

- Bash, How to assign a heredoc value to a variable in Bash?
- Bash, The while loop
- Bash, Length of string in bash
- Bash, Command Substitution
- FreeBSD, Man nc
- FreeBSD Forums, nc Server Not Disconnecting
- FreeBSD, Getting Started With tor Hidden Services on FreeBSD
- HTML, Coding An HTML 5 Layout From Scratch
- HTML, The Complete List of MIME Types
- HTML, W3C Markup Validation Service
- UNIX, Simple command line http server
- UNIX, One command line web server on port 80 using nc (netcat)
- UNIX, Faking Services using Netcat (For Testing Nagios)
- Wikipedia, Netcat: Setting up a one-shot webserver on port 8080 to present the content of a file