

Simple Socket Server in Bash?

Is there a way to quickly bind to a TCP port/ip address and simply print out all information to STDOUT? I have a simple debugging solution which writes things to 127.0.0.1:4444 and I'd like to be able to simply bind up a port from bash and print everything that comes across. Is there an easy way to do this?

linux

bash

sockets

tcp

related [unix.stackexchange.com/questions/49936/...](#) – [Ciro Santilli](#) 包子露宛 六四事件 法轮功 Sep 2 '16 at 22:20

4 Answers

```
$ nc -k -l 4444 > filename.out
```

see [nc\(1\)](#)

3 Any way to do it that doesn't require it to run in a loop, ie bind until killed? I'm repeatedly connecting and disconnecting to the socket and `nc` dies if I don't run it like this: `while true; do nc -l 4444; done` . – [Naftuli Kay](#) Jan 19 '11 at 18:43

2 Add `-k` option. – [Nikolai Fetissov](#) Jan 19 '11 at 18:45


On some distros you'll need to change it to 'nc -k -l -p 4444'. – [Rostislav Matl](#) Apr 9 '15 at 11:38

1 If you use `ncat` instead of `nc`, you can have multiple concurrent connections while using the exact same syntax. – [Sietse van der Molen](#) Sep 14 '15 at 2:56

1 yes, use `ncat` instead of `nc` (it comes bundled with `nmap` and it's a modern day incarnation of `nc`) – [Freedom_Ben](#) Jul 7 '16 at 18:06

Just because you asked how to do it in `bash` , though `netcat` answer is very valid:

```
$ exec 3<>/dev/tcp/127.0.0.1/4444
$ cat <&3
```

11 But that doesn't work for listening. I don't think its possible to listen using strictly bash – [Vijayender](#) Sep 5 '12 at 11:35 

6 This solution indeed requires a listening server. Bash cannot do this by means of `/dev/tcp` as explained in [unix.stackexchange.com/a/49947/13746](#) – [xebeche](#) Jun 12 '13 at 21:07

1 Listening? That's what xinetd is for! :D – [Evi1M4chine](#) Jan 23 '16 at 23:10

That is working as you expecting:

```
nc -k -l 4444 | bash
```

and then you

```
echo "ls" >/dev/tcp/127.0.0.1/4444
```

then you see the listing performed by bash.

[A Brief Security Warning]

Of course if you leave a thing like this running on your computer, you have a wide open gateway for all kinds of attacks because commands can be sent from any user account on any host in your network. This implements no security (authentication, identification) whatsoever and sends all transmitted commands unencrypted over the network, so it can very easily be abused.

Adding an answer using `ncat` that [@Freedom_Ben](#) alluded to:

```
ncat -k -l 127.0.0.1 4444
```

and explanation of options from `man ncat`:

<code>-k, --keep-open</code>	Accept multiple connections in listen mode
<code>-l, --listen</code>	Bind and listen for incoming connections

answered Aug 21 '17 at 20:45

 Kilokahn

1,144

13

35

`ncat` comes with `nmap` and supports concurrent connections, while the legacy `nc` command does not. – [Serge Stroobandt](#) Jan 12 at 17:32