THE GEEK STUFF
Linux | DB | Open Source | Web

eBook on Scaling Promethe
The ultimate guide to scaling prometheus collection issues.

≡ Menu

- Home
- Free eBook
- Start Here
- Contact
- About

8 Practical Linux Netcat NC Command Examples

by Himanshu Arora on April 23, 2012

Like 2 Tweet

Netcat or nc is a networking utility for debugging and investigating the network.

This utility can be used for creating TCP/UDP connections and investigating them. The biggest use of this utility is in the scripts where we need to deal with TCP/UDP sockets.

In this article we will learn about the netcat command by some practical examples.

1. Netcat in a Server-Client Architecture

The netcat utility can be run in the server mode on a specified port listening for incoming connections.

```
$ nc -l 2389
```

Also, it can be used in client mode trying to connect on the port(2389) just opened

```
$ nc localhost 2389
```

Now, if we write some text at the client side, it reaches the server side. Here is the proof :

```
$ nc localhost 2389
HI, server
```

On the terminal where server is running :




```
$ nc -l 2389
HI, server
```

So we see that netcat utility can be used in the client server socket communication.

2. Use Netcat to Transfer Files

The netcat utility can also be used to transfer files. At the client side, suppose we have a file named ‘testfile’ containing :

```
$ cat testfile
hello test
```

and at the server side we have an empty file ‘test’

Now, we run the server as :

```
$ nc -l 2389 > test
```

and run the client as :

```
cat testfile | nc localhost 2389
```

Now, when we see the ‘test’ file at the server end, we see :

```
$ cat test
hello test
```

So we see that the file data was transfered from client to server.

3. Netcat Supports Timeouts

There are cases when we do not want a connection to remain open forever. In that case, through ‘-w’ switch we can specify the timeout in a connection. So after the seconds specified along with -w flag, the connection between the client and server is terminated.

Server :

```
nc -l 2389
```

Client :

```
$ nc -w 10 localhost 2389
```

The connection above would be terminated after 10 seconds.

NOTE : Do not use the -w flag with -l flag at the server side as in that case -w flag causes no effect and hence the connection remains open forever.

4. Netcat Supports IPV6 Connectivity

The flag -4 or -6 specifies that netcat utility should use which type of addresses. -4 forces nc to use IPV4 address while -6 forces nc to use IPV6 address.

Server :

```
$ nc -4 -l 2389
```

Client :

```
$ nc -4 localhost 2389
```

Now, if we run the netstat command, we see :

```
$ netstat | grep 2389
tcp        0      0 localhost:2389          localhost:50851         ESTABLISHED
tcp        0      0 localhost:50851         localhost:2389          ESTABLISHED
```

The first field in the above output would contain a postfix '6' in case the IPV6 addresses are being used. Since in this case it is not, so a connection between server and client is established using IPV4 addresses.

Now, If we force nc to use IPV6 addresses

Server :

```
$ nc -6 -l 2389
```

Client :

```
$ nc -6 localhost 2389
```

Now, if we run the netstat command, we see :

```
$ netstat | grep 2389
tcp6       0      0 localhost:2389          localhost:33234         ESTABLISHED
tcp6       0      0 localhost:33234         localhost:2389          ESTABLISHED
```

So now a postfix '6' with 'tcp' shows that nc is now using IPV6 addresses.

5. Disable Reading from STDIN in Netcat

This functionality can be achieved by using the flag -d. In the following example, we used this flag at the client side.

Server :

```
$ nc -l 2389
```

Client :

```
$ nc -d localhost 2389
Hi
```

The text 'Hi' will not be sent to the server end as using -d option the read from stdin has been disabled.

6. Force Netcat Server to Stay Up

If the netcat client is connected to the server and then after sometime the client is disconnected then normally netcat server also terminates.

Server :

```
$ nc -l 2389
```

Client :

```
$ nc localhost 2389
^C
```

Server :

```
$ nc -l 2389
$
```

So, in the above example we see that as soon as the client got disconnected the server was also terminated.

This behavior can be controlled by using the -k flag at the server side to force the server to stay up even after the client has disconnected.

Server :

```
$ nc -k -l 2389
```

Client :

```
$ nc localhost 2389
^C
```

Server :

```
$ nc -k -l 2389
```

So we see that by using the -k option the server remains up even if the client got disconnected.

7. Configure Netcat Client to Stay Up after EOF

Netcat client can be configured to stay up after EOF is received. In a normal scenario, if the nc client receives an EOF character then it terminates immediately but this behavior can also be controlled if the -q flag is used. This flag expects a number which depicts number of seconds to wait before client terminates (after receiving EOF)

Client should be started like :

```
nc  -q 5  localhost 2389
```

Now if the client ever receives an EOF then it will wait for 5 seconds before terminating.

8. Use Netcat with UDP Protocol

By default all the sockets that nc utility creates are TCP protocols but this utility also works with UDP protocol. To enable UDP protocol the -u flag is used.

Server :

```
$ nc -4 -u -l 2389
```

Client :

```
$ nc -4 -u localhost 2389
```

Now, both the server and client are configured to use UDP protocol. This can be confirmed by the following netstat command. So we see that this connection is now using the UDP protocol.

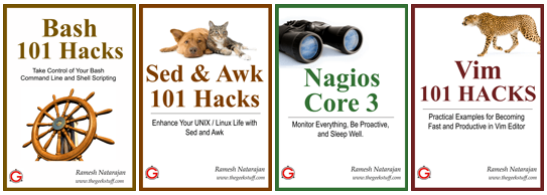
```
$ netstat | grep 2389
udp        0      0 localhost:42634         localhost:2389          ESTABLISHED
```

Tweet Like 2 > Add your comment

If you enjoyed this article, you might also like..

1. 50 Linux Sysadmin Tutorials
2. 50 Most Frequently Used Linux Commands (With Examples)
3. Top 25 Best Linux Performance Monitoring and Debugging Tools
4. Mommy, I found it! – 15 Practical Linux Find Command Examples
5. Linux 101 Hacks 2nd Edition eBook Free

- Awk Introduction – 7 Awk Print Examples
- Advanced Sed Substitution Examples
- 8 Essential Vim Editor Navigation Fundamentals
- 25 Most Frequently Used Linux IPTables Rules Examples
- Turbocharge PuTTY with 12 Powerful Add-Ons



{ 18 comments… add one }

- Pierre B. April 23, 2012, 1:32 am

  Thanks for this overview. I just encountered an “nc” snippet in a script i have to debug, and as i never used it myself i guess this page will help me with this task.

  Thank you Himanshu

  Link
- Himanshu April 23, 2012, 2:34 am

  @Pierre B
  Thanks!!

  Link
- Saurabh April 23, 2012, 3:31 am

  Nice post. Thanks ☺

  Link
- rinku April 23, 2012, 4:26 am

  Great …….

  Link
- Shaheem April 23, 2012, 2:11 pm

  Always great articles on this site. Is there any way to use netcat for multicasting udp or is there a different util for that? And for either answer if there is, will you be covering examples of it please?

  Link
- shark巨菜 April 24, 2012, 6:15 am

  A good introductory article. ☺

  Link
- Mohan April 24, 2012, 12:45 pm

  Thanks, its very useful for me.

  Link
- Anonymous April 26, 2012, 10:39 am

  Typo in section 3:
  NOTE : Do not use the -w flag with -l flag at the server side as in that case -w flag causes no AFFECT and hence the connection remains open forever.

  should read
  NOTE : Do not use the -w flag with -l flag at the server side as in that case -w flag causes no EFFECT and hence the connection remains open forever.

  Otherwise, very pleasant read and well structured post. Thx

  Link
- Ramesh Natarajan April 26, 2012, 11:42 pm

  @Anonymous,

  Thanks for pointing it out. It is fixed.

  Link
- Tanky Woo January 13, 2013, 8:04 am

  Hello, in the first one : Netcat in a Server-Client Architecture

  nc -l a_port doesn’t listen a_port. You should use nc -l -p a_port

  Link
- SomeGuy March 13, 2013, 7:10 pm

  Tanky Woo: There are at least two different versions of netcat around (“traditional” and “openbsd”, both available in Ubuntu etc). The traditional one requires -p to specify the port, while it’s optional in the openbsd version. At some point, some version (some older openbsd version?) complained if you used -p.

  Link
- James May 22, 2013, 6:25 am

  Hi guys,
  I’m trying to understand NC a little better and this is a great tutorial! I’m trying to redirect a com port via udp, but starting out, I’m just testing a server that stays alive.

  The following works fine, but if I disconnect the client, The server stays live, but when I reconnect again, no data is passed and the client drops back to the shell after attempting to transmit. Any thoughts?

  Sever:
  nc -k -l -4 -u 2221

  Client
  nc -4 -u localhost 2221

Link

- unggul August 11, 2013, 4:13 am

  wow great post !
  Hello im new in linux programming..i'm trying to send streaming packet data from my raspberry pi.
  My packet data is stored on /home/pi/file , then i want to send that data to my android phone..

  here is my code

  cat /home/pi/file | awk -F'\t' 'END{print}' | nc -l 8080

  is that right ?

  because on my android phone when receive the packet data is success,but sometimes jammed .
  Thanks for advance !

  Link
- Joe Antkowiak December 29, 2014, 4:43 pm

  UDP doesn't work the way you have it defined here, there is no such thing as a UDP "connection"

  UDP is connection-less and only listens for data packets on a given port, and NC behaves quite differently with UDP than it does with TCP because of this.

  Your output showing a "UDP Connection" appears to be fabricated. Netstat will never show you an "established state" for anything UDP because there are no connections to establish. It won't even show you client addresses because there are no connections, just packets coming in on a given port from any source.

  Link
- Marius Constantinescu September 25, 2015, 7:53 am

  very good article.

  thank you!

  Link
- Brock Judd April 13, 2016, 9:48 pm

  Thanks for a very useful page and examples.

  Link
- PK April 5, 2017, 6:12 am

  Thanks for these simple clear examples…

  Link
- Pankaj Kumar Saini May 6, 2017, 8:13 am

  Thanks Sir. Very nice article. But I had a problem:
  When I used IP6 to connect to Server my client or Client side terminal returned me an ERROR:
  nc: getaddrinfo: Name or service not known

  I googled the problem but could not found satisfactory answers. It will be very helpful if you can reply the solution. Thanks very much.

  Link

Leave a Comment

Name

Email

Website

Comment

Submit

Notify me of followup comments via e-mail

Next post: How to Create Linux Proc Files in C Program using LKM

Previous post: 20 Awesome Google Chrome Browser Tips and Tricks

RSS | Email | Twitter | Facebook | Google+

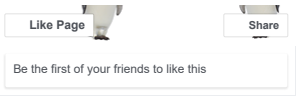
Google Custom Search Search



EBOOKS

- Free Linux 101 Hacks 2nd Edition eBook - Practical Examples to Build a Strong Foundation in Linux
- Bash 101 Hacks eBook - Take Control of Your Bash Command Line and Shell Scripting
- Sed and Awk 101 Hacks eBook - Enhance Your UNIX / Linux Life with Sed and Awk
- Vim 101 Hacks eBook - Practical Examples for Becoming Fast and Productive in Vim Editor
- Nagios Core 3 eBook - Monitor Everything, Be Proactive, and Sleep Well





POPULAR POSTS

- 15 Essential Accessories for Your Nikon or Canon DSLR Camera
- 12 Amazing and Essential Linux Books To Enrich Your Brain and Library
- 50 UNIX / Linux Sysadmin Tutorials
- 50 Most Frequently Used UNIX / Linux Commands (With Examples)
- How To Be Productive and Get Things Done Using GTD
- 30 Things To Do When you are Bored and have a Computer
- Linux Directory Structure (File System Structure) Explained with Examples
- Linux Crontab: 15 Awesome Cron Job Examples
- Get a Grip on the Grep! – 15 Practical Grep Command Examples
- Unix LS Command: 15 Practical Examples
- 15 Examples To Master Linux Command Line History
- Top 10 Open Source Bug Tracking System
- Vi and Vim Macro Tutorial: How To Record and Play
- Mommy, I found it! -- 15 Practical Linux Find Command Examples
- 15 Awesome Gmail Tips and Tricks
- 15 Awesome Google Search Tips and Tricks
- RAID 0, RAID 1, RAID 5, RAID 10 Explained with Diagrams
- Can You Top This? 15 Practical Linux Top Command Examples
- Top 5 Best System Monitoring Tools
- Top 5 Best Linux OS Distributions
- How To Monitor Remote Linux Host using Nagios 3.0
- Awk Introduction Tutorial – 7 Awk Print Examples
- How to Backup Linux? 15 rsync Command Examples
- The Ultimate Wget Download Guide With 15 Awesome Examples
- Top 5 Best Linux Text Editors
- Packet Analyzer: 15 TCPDUMP Command Examples
- The Ultimate Bash Array Tutorial with 15 Examples
- 3 Steps to Perform SSH Login Without Password Using ssh-keygen & ssh-copy-id
- Unix Sed Tutorial: Advanced Sed Substitution Examples
- UNIX / Linux: 10 Netstat Command Examples
- The Ultimate Guide for Creating Strong Passwords
- 6 Steps to Secure Your Home Wireless Network
- Turbocharge PuTTY with 12 Powerful Add-Ons

CATEGORIES

- Linux Tutorials
- Vim Editor
- Sed Scripting
- Awk Scripting
- Bash Shell Scripting
- Nagios Monitoring
- OpenSSH
- IPTables Firewall
- Apache Web Server
- MySQL Database
- Perl Programming
- Google Tutorials
- Ubuntu Tutorials
- PostgreSQL DB
- Hello World Examples
- C Programming
- C++ Programming
- DELL Server Tutorials
- Oracle Database
- VMware Tutorials

About The Geek Stuff


My name is **Ramesh Natarajan**. I will be posting instruction guides, how-to, troubleshooting tips and tricks on Linux, database, hardware, security and web. My focus is to write articles that will either teach you or help you resolve a problem. Read more about Ramesh Natarajan and the blog.

Contact Us

Email Me : Use this Contact Form to get in touch me with your comments, questions or suggestions about this site. You can also simply drop me a line to say hello!.

Follow us on Google+

Follow us on Twitter

Become a fan on Facebook

Support Us

Support this blog by purchasing one of my ebooks.

Bash 101 Hacks eBook

Sed and Awk 101 Hacks eBook

Vim 101 Hacks eBook

Nagios Core 3 eBook

Copyright © 2008–2018 Ramesh Natarajan. All rights reserved | Terms of Service