Linux netstat command explained with 10 examples (Real world Linux) https://www.ibm.com/developerworks/community/blogs/58e72888-6340-46ac-b488-d31aa4058e9c/entry/linux_netstat_command_explained_wi...

IBM | Sign in | Register dW

IBM Technical topics Evaluation software Community Events Search developerWorks

Profiles Communities Apps English

Blogs My Blogs Public Blogs My Updates This Blog Search

Log in
to participate

Real world Linux

Overview
Recent Updates
Members
Forums
Feeds
Bookmarks
Blog
Wiki

Linux netstat command explained with 10 examples

Himanshuz.chd | *Aug 25 2012* | *Comment (1)* | *Visits (181165)*

G+ Like 11 Share Tweet

The netstat command in Linux is a very useful tool when dealing with networking issues. This command is capable of producing information related to network connections, routing tables, interface statistics etc. This utility also helps the network administrators to keep an eye on the invalid or suspicious network connections. In this article we will understand the basics of this command using some practical examples. The syntax of this command is :

```
netstat [options]...
```

1. Display routing information maintained by kernel

This information can be retrieved using the -r option along with this command.

Consider the following example :

```
$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt
Iface
192.168.1.0     *               255.255.255.0   U         0 0          0
wlan0
link-local      *               255.255.0.0     U         0 0          0
wlan0
default         192.168.1.1     0.0.0.0         UG        0 0          0
wlan0
```

So we see that kernel routing table information was displayed using the -r option. The flag 'U' indicates that this entry is up while the flag 'G' indicates that this entry is not a direct entry i.e. the destination indicated in this route entry is not on the same network. A list of flags is given below :

A Receive all multicast at this interface.
B OK broadcast.
D Debugging ON.
M Promiscuous Mode.
O No ARP at this interface.
P P2P connection at this interface.
R Interface is running.
U Interface is up.
G Not a direct entry.

2. Display multicast group membership information

This information is displayed for both IPv4 and IPv6 and can be retrieved using -g option with this command.

Consider the following example :

```
$ netstat -g
IPv6/IPv4 Group Memberships
Interface       RefCnt Group
--------------- ------ ---------------------
lo              1      all-systems.mcast.net
eth0            1      all-systems.mcast.net
wlan0           1      224.0.0.251
wlan0           1      all-systems.mcast.net
lo              1      ip6-allnodes
eth0            1      ip6-allnodes
wlan0           1      ff02::1:ff20:3a8e
wlan0           1      ip6-allnodes
pan0            1      ip6-allnodes
```

So we see that the multicast information was displayed in the above output.

3. Display information related to all network interfaces

This is made possible using the -i option along with this command.

Consider the following example :

```
$ netstat -i
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500 0         0      0      0 0             0      0      0
0 BMU
lo        16436 0        44      0      0 0            44      0      0
0 LRU
wlan0      1500 0    166164      0      0 0        152434      0      0
0 BMRU
```

So we see that all the network information related to individual interfaces was displayed in the output. The

About this blog

Linux is alive and very well, doing real work in the real world. Share your successes, your questions and your ideas with authors and others in the developerWorks community.

Related posts

IBM Sterling B2B Int...
Updated July 10 0 0
How to Focus in Your...
Updated July 11 0 0
Configuring network ...
Updated Apr 19 2 0
Freedom Network Reve...
Updated Mar 16 0 0
IBM DataPower Gatewa...
Updated Feb 14 8 0

Tags

Find a Tag

0x80 12.04 13.04 2012 2013 allocation alpha1 argument arguments basics blktrace block bonzini boot buffer built byte c c/c++ call calls canonical checksum choice cloud command commandline commands controversy development editor file filesystem getopt kernel line linux new open os process stack storage system tips top tricks ubuntu user vim

Cloud List

RX and TX columns are described as follows :

RX-OK : Correct packets received on this interface.
RX-ERR : Incorrect packets received on this interface
RX-DRP : Packets that were dropped at this interface.
RX-OVR : Packets that this interface was unable to receive.

Similar definition is for the **TX** columns that describe the transmitted packets.

4. Display summary statistics for each protocol

This is very handy information that netstat command provides. This information can be retrieved by using -s option with this command.

Consider the following example :

```
$ netstat -s
Ip:
    167813 total packets received
    1 with invalid addresses
    0 forwarded
    0 incoming packets discarded
    166864 incoming packets delivered
    153028 requests sent out
Icmp:
    12 ICMP messages received
    0 input ICMP message failed.
    ICMP input histogram:
        destination unreachable: 12
    12 ICMP messages sent
    0 ICMP messages failed
    ICMP output histogram:
        destination unreachable: 12
IcmpMsg:
        InType3: 12
        OutType3: 12
Tcp:
    3270 active connections openings
    0 passive connection openings
    11 failed connection attempts
    279 connection resets received
    2 connections established
    158262 segments received
    145989 segments send out
    477 segments retransmited
    0 bad segments received.
    1938 resets sent
Udp:
    5418 packets received
    12 packets to unknown port received.
    0 packet receive errors
    5387 packets sent
UdpLite:
TcpExt:
    52 packets pruned from receive queue because of socket buffer overrun
    1661 TCP sockets finished time wait in fast timer
    2 time wait sockets recycled by time stamp
    5 packets rejects in established connections because of timestamp
    3733 delayed acks sent
    1 delayed acks further delayed because of locked socket
    Quick ack mode was activated 890 times
    384 packets directly queued to recvmsg prequeue.
    210504 bytes directly received in process context from prequeue
    83445 packet headers predicted
    153 packets header predicted and directly queued to user
    8241 acknowledgments not containing data payload received
    1732 predicted acknowledgments
    3 congestion windows recovered without slow start by DSACK
    203 congestion windows recovered without slow start after partial ack
    1 timeouts after reno fast retransmit
    9 timeouts after SACK recovery
    2 timeouts in loss state
    9 retransmits in slow start
    428 other TCP timeouts
    1782 packets collapsed in receive queue due to low socket buffer
    861 DSACKs sent for old packets
    22 DSACKs sent for out of order packets
    276 DSACKs received
    407 connections reset due to unexpected data
    272 connections reset due to early user close
    4 connections aborted due to timeout
    TCPDSACKIgnoredOld: 128
    TCPDSACKIgnoredNoUndo: 31
    TCPSackShiftFallback: 2
IpExt:
    InMcastPkts: 2370
    OutMcastPkts: 1199
    InBcastPkts: 2270
    OutBcastPkts: 1331
    InOctets: 194805011
    OutOctets: 15947915
    InMcastOctets: 74161
    OutMcastOctets: 41385
    InBcastOctets: 477074
    OutBcastOctets: 194151
```

So we see that vital statistical information related to each protocol was displayed in the output.

5. Monitor continuously

Yes, netstat command provides an option -c using which any type of information can be monitored continuously. Here continuously means that same information would be fetched again and again after each second and the netstat output will grow until you choose to stop the command.

Here is an example where the interface information can be monitored continuously :

```
$ netstat -ic
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500 0         0      0      0 0             0      0      0
0 BMU
lo        16436 0        44      0      0 0            44      0      0
0 LRU
wlan0      1500 0    167000      0      0 0        153174      0      0
0 BMRU
```

```
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500 0         0      0      0 0             0      0      0
0 BMU
lo        16436 0        44      0      0 0            44      0      0
0 LRU
wlan0      1500 0    167000      0      0 0        153174      0      0
0 BMRU
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500 0         0      0      0 0             0      0      0
0 BMU
lo        16436 0        44      0      0 0            44      0      0
0 LRU
wlan0      1500 0    167000      0      0 0        153174      0      0
0 BMRU
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500 0         0      0      0 0             0      0      0
0 BMU
lo        16436 0        44      0      0 0            44      0      0
0 LRU
wlan0      1500 0    167000      0      0 0        153174      0      0
0 BMRU
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500 0         0      0      0 0             0      0      0
0 BMU
lo        16436 0        44      0      0 0            44      0      0
0 LRU
wlan0      1500 0    167000      0      0 0        153174      0      0
0 BMRU
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500 0         0      0      0 0             0      0      0
0 BMU
lo        16436 0        44      0      0 0            44      0      0
0 LRU
wlan0      1500 0    167000      0      0 0        153174      0      0
0 BMRU
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500 0         0      0      0 0             0      0      0
0 BMU
lo        16436 0        44      0      0 0            44      0      0
0 LRU
wlan0      1500 0    167000      0      0 0        153174      0      0
0 BMRU
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500 0         0      0      0 0             0      0      0
0 BMU
lo        16436 0        44      0      0 0            44      0      0
0 LRU
wlan0      1500 0    167001      0      0 0        153174      0      0
0 BMRU
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500 0         0      0      0 0             0      0      0
0 BMU
lo        16436 0        44      0      0 0            44      0      0
0 LRU
wlan0      1500 0    167001      0      0 0        153175      0      0
0 BMRU
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500 0         0      0      0 0             0      0      0
0 BMU
lo        16436 0        44      0      0 0            44      0      0
0 LRU
wlan0      1500 0    167001      0      0 0        153175      0      0
0 BMRU
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500 0         0      0      0 0             0      0      0
0 BMU
lo        16436 0        44      0      0 0            44      0      0
0 LRU
wlan0      1500 0    167001      0      0 0        153175      0      0
0 BMRU
^C
$
```

So we see that the interface information (using -i) was displayed continuously again and again using -c option. The figure change (highlighted in **bold**) in the output above gives us an idea how -c option is useful to see updates in statistics in real time.

6. Display extra information

Apart from the information that netstat produces in output, Extra information can be produced in output using -e option.

Consider an example below :

```
$ netstat -e
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
User       Inode
tcp        0      0 himanshu-laptop.l:46096 sjc-not16.sjc.dropb:www
ESTABLISHED himanshu   88185
tcp       38      0 himanshu-laptop.l:40156 v-d-1a.sjc.dropbo:https
CLOSE_WAIT  himanshu   88182
tcp       38      0 himanshu-laptop.l:54501 v-client-5a.sjc.d:https
CLOSE_WAIT  himanshu   247035
tcp       38      0 himanshu-laptop.l:60738 v-client-2b.sjc.d:https
CLOSE_WAIT  himanshu   10991
tcp        0      0 himanshu-laptop.l:59610 del01s05-in-f22.1:https
ESTABLISHED himanshu   186169
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    3273     @/org/kernel/ude
v/udevd
unix  20     [ ]         DGRAM                    4787     /dev/log
unix  3      [ ]         STREAM     CONNECTED     206978   @/tmp/dbus-VwQ8G
S3QiP
unix  3      [ ]         STREAM     CONNECTED     206977
unix  3      [ ]         STREAM     CONNECTED     206943   @/tmp/dbus-VwQ8G
S3QiP
unix  3      [ ]         STREAM     CONNECTED     206942
unix  3      [ ]         STREAM     CONNECTED     206941   @/tmp/.ICE-unix/
1543
unix  3      [ ]         STREAM     CONNECTED     206940
unix  3      [ ]         STREAM     CONNECTED     206939   @/tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     206938
unix  3      [ ]         STREAM     CONNECTED     206937   /tmp/orbit-himan
shu/linc-dcf-0-427340219d277
unix  3      [ ]         STREAM     CONNECTED     206936
unix  3      [ ]         STREAM     CONNECTED     206933   /tmp/orbit-himan
shu/linc-630-0-480531b88e2fc
...
...
...
```

So we see that lots of extra information related to internet connections (like user, Inode etc) was produced in the output.

7. Display network timer related information

This type of information can be produced in output using -o option with this command.

Consider the following example :

```
$ netstat -o
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Timer
tcp        0      0 himanshu-laptop.l:46096 sjc-not16.sjc.dropb:www
ESTABLISHED off (0.00/0/0)
tcp       38      0 himanshu-laptop.l:40156 v-d-1a.sjc.dropbo:https
CLOSE_WAIT  off (0.00/0/0)
tcp       38      0 himanshu-laptop.l:54501 v-client-5a.sjc.d:https
CLOSE_WAIT  off (0.00/0/0)
tcp       38      0 himanshu-laptop.l:60738 v-client-2b.sjc.d:https
CLOSE_WAIT  off (0.00/0/0)
tcp        0      0 himanshu-laptop.l:59610 del01s05-in-f22.1:https
ESTABLISHED off (0.00/0/0)
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    3273     @/org/kernel/ude
v/udevd
unix  20     [ ]         DGRAM                    4787     /dev/log
unix  3      [ ]         STREAM     CONNECTED     206978   @/tmp/dbus-VwQ8G
S3QiP
unix  3      [ ]         STREAM     CONNECTED     206977
unix  3      [ ]         STREAM     CONNECTED     206943   @/tmp/dbus-VwQ8G
S3QiP
unix  3      [ ]         STREAM     CONNECTED     206942
...
...
...
```

So we see that the timer related information (highlighted in bold) was produced in the output.

8. Display the PID of the program using socket

The PID of the program using a particular socket can be produced in the output using the option -p with this command.

Consider the following example :

```
$ netstat -p
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0    195 himanshu-laptop.l:46096 sjc-not16.sjc.dropb:www
ESTABLISHED 1643/dropbox
tcp       38      0 himanshu-laptop.l:40156 v-d-1a.sjc.dropbo:https
CLOSE_WAIT  1643/dropbox
tcp       38      0 himanshu-laptop.l:54501 v-client-5a.sjc.d:https
CLOSE_WAIT  1643/dropbox
tcp       38      0 himanshu-laptop.l:60738 v-client-2b.sjc.d:https
CLOSE_WAIT  1643/dropbox
tcp        0      0 himanshu-laptop.l:59610 del01s05-in-f22.1:https
ESTABLISHED 1887/firefox
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   PID/Program name
Path
unix  2      [ ]         DGRAM                    3273     -
@/org/kernel/udev/udevd
unix  20     [ ]         DGRAM                    4787     -
/dev/log
unix  3      [ ]         STREAM     CONNECTED     206978   1581/dbus-daemon
@/tmp/dbus-VwQ8GS3QiP
unix  3      [ ]         STREAM     CONNECTED     206977   1627/metacity
...
...
...
```

As suggested by the highlighted portion in the output, the PID related information was produced using -p option.

9. Show only listening sockets

This can be made possible by using the -l option with this command.

Consider the following example :

```
$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:ipp           *:*                     LISTEN
tcp        0      0 *:17500                 *:*                     LISTEN
tcp6       0      0 [::]:netbios-ssn        [::]:*                  LISTEN
tcp6       0      0 localhost:ipp           [::]:*                  LISTEN
tcp6       0      0 [::]:microsoft-ds       [::]:*                  LISTEN
...
...
...
```

So we see that only those sockets whose state is LISTEN are produced in the output.

10. Show routing information from route cache

Information from route cache is produced in the output using -C option with this command.

Consider the following example :

```
$ netstat -C
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 himanshu-laptop.l:46096 sjc-not16.sjc.dropb:www
ESTABLISHED
tcp       38      0 himanshu-laptop.l:40156 v-d-1a.sjc.dropbo:https
CLOSE_WAIT
tcp       38      0 himanshu-laptop.l:54501 v-client-5a.sjc.d:https
CLOSE_WAIT
tcp       38      0 himanshu-laptop.l:60738 v-client-2b.sjc.d:https
CLOSE_WAIT
tcp        0      0 himanshu-laptop.l:59610 del01s05-in-f22.1:https
ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    3273     @/org/kernel/ude
v/udevd
unix  20     [ ]         DGRAM                    4787     /dev/log
unix  3      [ ]         STREAM     CONNECTED     206978   @/tmp/dbus-VwQ8G
S3QiP
unix  3      [ ]         STREAM     CONNECTED     206977
unix  3      [ ]         STREAM     CONNECTED     206943   @/tmp/dbus-VwQ8G
S3QiP
unix  3      [ ]         STREAM     CONNECTED     206942
...
...
...
```

The output above is produced using the information from route cache.

Tags: *network netstat route*

Add a Comment More Actions

Comments (1) ***Add a Comment*** ***More Actions***

Coldking123 *commented Oct 30 2014* *Comment Permalink*

whenever i use any of these commands on my Ubuntu terminal. I just get a long list of sockets (or something like that). starting with unix 2/3. what command do i use to make it to where i can view information as it is portrayed above?

Previous Entry Main Next Entry

About
Help
Contact us
Submit content

Feeds
Newsletters
Follow
Like

Report abuse
Terms of use
Third party notice
IBM privacy
IBM accessibility

Faculty
Students
Business Partners

Select a language:
English
中文
日本語
Русский
Português (Brasil)
Español
Việt