Find number of unique IP’s and active connections to Web server

September 25, 2014 [How to](https://www.blackmoreops.com/how-to/), [Linux](https://www.blackmoreops.com/linux/), [Linux Administration](https://www.blackmoreops.com/administration/), [Monitoring](https://www.blackmoreops.com/monitoring/) [5 Comments](https://www.blackmoreops.com/2014/09/25/find-number-of-unique-ips-active-connections-to-web-server/#comments)

* [Facebook](http://www.facebook.com/sharer.php?u=https://www.blackmoreops.com/?p=2786)
* [Twitter](https://twitter.com/intent/tweet?text=Find+number+of+unique+IP%E2%80%99s+and+active+connections+to+Web+server%20via%20%40blackmoreops&url=https://www.blackmoreops.com/?p=2786)
* [Google +](https://plusone.google.com/_/+1/confirm?hl=en&url=https://www.blackmoreops.com/?p=2786&name=Find+number+of+unique+IP%E2%80%99s+and+active+connections+to+Web+server)

In computing, netstat (**net**work **stat**istics) is a command-line tool that displays network connections (both incoming and outgoing), routing tables, and a number of network interface (network interface controller or software-defined network interface) and network protocol statistics. It is available on Unix-like operating systems including OS X, Linux, Solaris, and BSD, and is available on Windows NT-based operating systems including Windows XP, Windows Vista, Windows 7 and Windows 8.[](http://www.blackmoreops.com/wp-content/uploads/2014/09/Find-number-of-active-connections-to-Web-server-blackMORE-Ops.jpg)

It is used for finding problems in the network and to determine the amount of traffic on the network as a performance measurement.

netstat is the most frequent tool used for monitoring network connections on a Linux servers. netstat returns a variety of information on active connections such as their current status, what hosts are involved, and which programs are involved. You can also see information about the routing table and even get statistics on your network interfaces. netstat is a good all-around utility and it is an essential tool for the Linux administrators.

If you just type netstat, it would display a long list of information that’s usually more than you want to go through at any given time. The trick is that how to keeping the information useful and what you’re looking for and how to tell netstat to only display that information.

Users can also use man netstat command to get detailed netstat help and manual where there are lots of configurable options and flags to get meaningful lists and results.

The Holy Grail of netstat – My way

My requirements are simple:

1. I want to see all users connected to my server
2. I only want to see users on port 80 (http)
3. I want to see total number of connections per IP
4. I don’t want to see my own server’s IP in the output
5. I want to sort them Max to Min.

Following commands will do just that. I am using the usual netstatty things. I will break down each command as I go:

Step 1: Show active connections – TCP and UDP Internet  (w/o servers)

First of all lets just check the active TCP and UDP connections.

netstat -ntu

-ntu = TCP and UDP connections in numeric order (w/o servers).

Step 2: Show all active connections – (/w servers and established)

Well, that’s good. But how do I know what ports I am listening to? I need that cause I want to check that my server is not listening to any funky ports. If it does, then I got either mis-configured services or my server is compromised.

So I will just add ‘ -a ‘ flag in this command.

netstat -antu

-antu = All TCP and UDP connections in numeric order (with servers and established)

I’ve seen way too many guides where people go an use ‘ plan ‘ flag .. ‘ -a ‘ already includes everything, ‘ -l ‘ is not required.

Step 3: Show all active connections to Web server

Now that I have a decent output, I can start working on it. In this particular output, I got all sorts active TCP and UDP Internet connections on every open port. I want to narrow it down to port 80 only (HTTP Port). I will use grep to narrow down on that.

netstat -antu | grep :80

Nice, now I see active TCP and UDP Internet connections on port 80 only. (in fact it will be all TCP as HTTP is a TCP service). You can use netstat -anu to list UDP connections.

Step 4: Show all active connections to Web server excluding self IP’s

So far I managed to list all active TCP and UDP connections and filter my results for port 80. But I can see few entries like following:

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN

tcp 0 0 :::80 :::\* LISTEN

I don’t want to exclude that as it is just loopback address. I am more interested to see foreign IP’s connected to my website. So I will just use grep -v to exclude these connections.

netstat -antu | grep :80 | grep -v LISTEN

>> Sample output

root@kali [~]# netstat -antu | grep :80 | grep -v LISTEN

tcp 0 0 127.0.0.1:80 202.28.177.61:59818 SYN\_RECV

tcp 0 0 127.0.0.1:80 202.28.177.61:40560 SYN\_RECV

tcp 0 0 127.0.0.1:80 202.28.177.61:50523 SYN\_RECV

tcp 0 0 127.0.0.1:80 202.28.177.61:55520 SYN\_RECV

tcp 0 0 127.0.0.1:80 68.204.187.5:50420 TIME\_WAIT

tcp 0 0 127.0.0.1:80 202.28.177.61:41423 TIME\_WAIT

tcp 0 0 127.0.0.1:80 24.232.175.224:45254 TIME\_WAIT

tcp 0 0 127.0.0.1:80 68.204.187.5:50465 TIME\_WAIT

tcp 0 0 127.0.0.1:80 202.28.177.61:43653 TIME\_WAIT

tcp 0 0 127.0.0.1:80 202.28.177.61:33851 TIME\_WAIT

Step 5: Show all active connections to Web server – IP: Port only

Now I want to do more. I want to exclude everything and only list foreign IP addresses. I need some sort of filter. awk is exactly the tool I need to use here. But how do I know which field to filter?

Let’s look at a sample output from previous command.

tcp 0 0 127.0.0.1:80 202.28.177.61:59818 SYN\_RECV

In the above line, I am only interested on the foreign IP 202.28.177.61. awk has the capability to do data extraction. Let’s count the fields.

(1)tcp (2)0 (3)0 (4)127.0.0.1:80 (5)202.28.177.61:59818 (6)SYN\_RECV

Note that, 127.0.0.1:80 and 202.28.177.61:59818 is counted as one field as they don’t have any spaces.

So, let’s try to list the 5th field which contains all the foreign IP addresses.

netstat -antu | grep :80 | grep -v LISTEN | awk '{print $5}'

>> Sample output

root@kali [~]# netstat -antu | grep :80 | grep -v LISTEN | awk '{print $5}'

113.173.16.70:26985

50.106.33.249:49292

113.173.16.70:26982

66.87.131.102:4370

113.173.16.70:26996

175.143.49.222:19709

66.87.131.102:4386

87.76.5.167:18296

23.24.156.251:57306

222.66.184.182:56000

Step 6: Show all active connections to Web server – IP’s only

The last output looks good to me. I managed to list only foreign addresses. But If you look closely, you will see same foreign IP is listed more than once on different ports. It is essentially the same user (or groups of users behind same proxy).

Let’s go one step further, I now want to see foreign IP’s only without their originating ports. I will be using cut for that. cut will allow me to cut every line at the first space character and take the first piece.

netstat -antu | grep :80 | grep -v LISTEN | awk '{print $5}' | cut -d: -f1

I am using -d: to let cut know that I am using the (:) colon character as the field delimiter.

By using -f1 I am telling cut to give me the first field (which is the IP address before the colon character).

>> Sample output

root@kali [~]# netstat -antu | grep :80 | grep -v LISTEN | awk '{print $5}' | cut -d: -f1

99.104.23.221

99.104.23.221

2.50.116.13

5.239.12.118

99.104.23.221

99.104.23.221

66.249.81.52

99.104.23.221

218.80.192.208

99.104.23.221

Step 7: Show all active connections to Web server – sorted and unique

Now the output contains only foreign IP addresses. We now need to sort them, and then pipe it to uniq command so that we are left with unique foreign IP in a sorted manner. I also want uniqcommand to count the number of connections per IP.

netstat -antu | grep :80 | grep -v LISTEN | awk '{print $5}' | cut -d: -f1 | sort | uniq -c

uniq -c will count total number of connections per IP.

>> Sample output

root@kali [~]# netstat -antu | grep :80 | grep -v LISTEN | awk '{print $5}' | cut -d: -f1 | sort | uniq -c

14 112.204.9.125

1 113.173.16.70

4 124.176.50.119

1 125.163.235.42

1 175.136.237.182

1 178.26.19.53

1 180.76.6.44

12 202.28.177.61

Step 8: Show all active connections to Web server – connection count

But hang on, uniq -c did count it and sorted them as per IP addresses. Let’s re-sort this list to show Max to Min number of connections per unique IP (foreign IP’s only).

netstat -antu | grep :80 | grep -v LISTEN | awk '{print $5}' | cut -d: -f1 | sort | uniq -c | sort -rn

>> Sample output

root@kali [~]# netstat -ntu | grep :80 | grep -v LISTEN | awk '{print $5}' | cut -d: -f1 | sort | uniq -c | sort -rn

44 92.45.59.194

37 121.52.153.215

35 212.14.31.37

28 76.177.223.119

28 37.228.105.222

27 182.186.25.222

8 13.1.89.53

6 127.0.0.1

Step 9: Show all active connections to Web server – exclude self IP’s

It’s all good so far. But I can see my own server’s IP in that list. (i.e. 127.0.0.1). If I can just exclude that, I will be left with Foreign IP’s only. I will use grep -v to exclude that line.

root@kali [~]# netstat -ntu | grep :80 | grep -v LISTEN | awk '{print $5}' | cut -d: -f1 | sort | uniq -c | sort -rn | grep -v 127.0.0.1

>> Sample output

root@kali [~]# netstat -ntu | grep :80 | grep -v LISTEN | awk '{print $5}' | cut -d: -f1 | sort | uniq -c | sort -rn | grep -v 127.0.0.1

45 108.168.173.83

36 174.45.176.119

35 70.39.187.230

34 175.144.92.225

24 139.195.97.102

19 69.142.186.97

18 150.101.190.51

12 50.150.246.167

Step 10: Final count of all active IP’s to Web server

This is now working just the way I wanted. Let’s do a final count of total number of unique IP’s connected to my server.

root@kali [~]# netstat -ntu | grep :80 | grep -v LISTEN | awk '{print $5}' | cut -d: -f1 | sort | uniq -c | sort -rn | grep -v 127.0.0.1 | wc -l

>> Sample output

root@kali [~]# netstat -ntu | grep :80 | grep -v LISTEN | awk '{print $5}' | cut -d: -f1 | sort | uniq -c | sort -rn | grep -v 127.0.0.1 | wc -l

768

wc -l does a line count..

It means I’ve got 768 unique IP’s connected right now.

Step 11: Final count of all active connections to Web server

Last but not the least, I want a total count of connections to port 80 from all foreign IP’s.

netstat -ntu | grep :80 | grep -v LISTEN | awk '{print $5}' | cut -d: -f1 | grep -v 127.0.0.1 | wc -l

>> Sample output

root@kali [~]# netstat -ntu | grep :80 | grep -v LISTEN | awk '{print $5}' | cut -d: -f1 | grep -v 127.0.0.1 | wc -l

1847

That means I’ve got 1847 connections opened right now from all external IP’s to port 80 on my server.

Conclusion

netstat is the most useful tool to detect and determine whether a server is under DoS or DDoS attack (Distributed Denial of Service). Whenever a client connects to a server via network, a connection is established and opened on the system. On a busy high load server, the number of connections connected to the server can be run into large amount till hundreds if not thousands. Find out and get a list of connections on the server by each node, client or IP address is useful for system scaling planning, and in most cases, detect and determine whether a web server is under DoS or DDoS attack (Distributed Denial of Service), where an IP sends large amount of connections to the server. To check connection numbers on the server, administrators and webmasters can make use of netstat command.

Do I think my way of using netstat is the best? Absolutely! Why? Cause it always worked for me.

I don’t know whether your sever is running on port 80 or if you’re getting DDoS on a different port (i.e. port 25 or 443), but the above command serves my purpose every time. There’s 10 different ways to skin a cat and I like my way. I’ve broken down my way of using netstat command to the last straw and once you follow it from top to bottom, you will get used to it and in time you will find your own perfect command.