



www.og150.com

 Follow @theog150

Iperf Bandwidth Performance Testing

TABLE OF CONTENTS

What Is 'Iperf'?	2
Why Would A Hacker Use 'Iperf'?	2
'Iperf' Demonstration.	3
Final Conclusions.	8



www.og150.com

Follow @theog150

What Is 'Iperf'?

In really simple terms, Iperf is a program that allows you to measure network throughput between two devices. Network throughput is the amount of bandwidth that was available during the test. To do this, Iperf identifies one device as the client and one device as the server, the client then initiates the throughput test using either TCP or UDP data streams (the client sends data to the server and calculates how long it takes to send the data, it can then work out the throughput that has been achieved). Once the test has completed, a summary of the throughput achieved is displayed. I really like this tool because it is simple to use, very lightweight and supports most platforms including Linux and Windows.

Why Would A Hacker Use 'Iperf'?

This is a legitimate question, after all Iperf is a legitimate network testing tool. To answer this, you need to think like a hacker..... Allow me to repeat the legitimate use for this tool: *Generate a significant amount of traffic and display the throughput between two devices.* Immediately, I think of the following:

- **DOS (Denial of Service) attack.** A malicious user could use Iperf to effectively choke a targets Internet bandwidth. For example, a malicious user deploys an OG150 (or similar security drop box) on the targets network infrastructure – let's say company Podo. They then launch Iperf to stream 100Mb/s of UDP traffic to a destination on the Internet. Depending on the Internet pipe used by company Podo, this could easily become saturated by the Iperf traffic and legitimate users on the network may lose Internet connectivity or access becomes so slow it is unusable. This can quickly lead to a DOS attack scenario.
- **Attack planning tool.** I will use the same scenario discussed previously, whereby a malicious user deploys an OG150 (or similar security drop box) on the targets network infrastructure – company Podo in my example. They then run Iperf to a destination on the Internet and Iperf calculates that there is 3Mb/s bi-directional throughput to the Internet from the deployed drop box. Using this information, the malicious user can more accurately plan further attacks. For example, my own testing shows that an 'Ai-Ball' wireless webcam connected to the OG150 requires 3.2Mb/s download and 80Kb/s upload throughput to stream live video and audio surveillance from the webcam installed on the targets site to my server on the Internet (Video and Audio with an image resolution VGA 640x480). If Iperf displays there is only 1Mb/s of bandwidth available, you will need to change the image resolution accordingly or consider other tools such as compression. In fact, if the available bandwidth is too small, live video/audio surveillance may be impossible.



www.og150.com

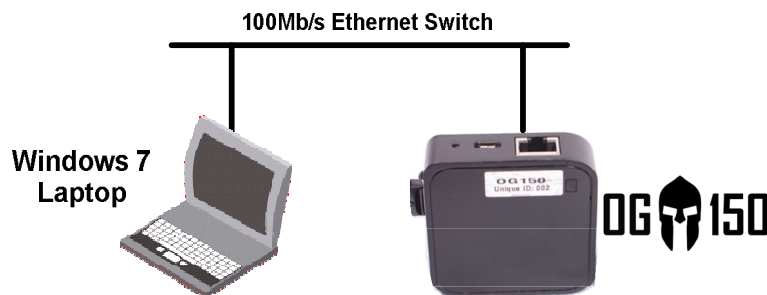
Follow @theog150

'Iperf' Demonstration.

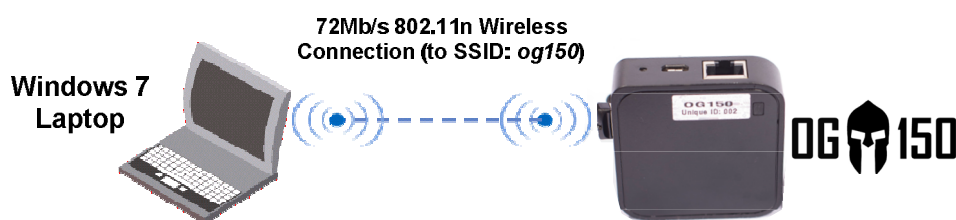
This tutorial will perform a wired throughput test and a wireless throughput test between an OG150 and a Windows 7 laptop. The OG150 comes with Iperf pre-installed, you can download the Windows version of Iperf from: <http://iperf.fr/>

The main focus of this tutorial is how to use Iperf, therefore I will keep the topologies very simple as shown in Screenshot 1 and Screenshot 2. I encourage you try different scenarios – for example I did an Iperf throughput test over a Reverse SSH tunnel between an OG150 and an AWS (Amazon Web Services) EC2 instance to see what kind of throughput there was between these devices.

Screenshot 1 – Iperf WIRED topology



Screenshot 2 – Iperf WIRELESS topology



First of all, confirm that Iperf is installed properly on the devices. Connect to the OG150 via an SSH session and enter 'iperf -v' – if Iperf is installed the version number will be displayed. As you can see in Screenshot 3 Iperf v2.0.5 is installed.

Screenshot 3 – OG150 Iperf version

```
root@OG150:~# iperf -v
iperf version 2.0.5 (08 Jul 2010) single threaded
root@OG150:~#
```



www.og150.com

Follow @theog150

On a Windows device, you need to download the Iperf program. Once this is downloaded, make a note of the directory where the file 'iperf.exe' resides. Open an MSDOS prompt and browse to the directory where 'iperf.exe' is located. To display the version of Iperf, use the same command 'iperf -v'. As you can see in Screenshot 4 Iperf v2.0.5 is installed.

Screenshot 4 – Windows 7 Iperf version

```
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\> iperf -v
iperf version 2.0.5 (08 Jul 2010) win32 threads
C:\>
```

Although there are a few features and options available with Iperf, there are two simple commands that will get you started.

- 1) Start the Iperf server with the command 'iperf -s'
- 2) Start the Iperf client with the command 'iperf -c x.x.x.x' where x.x.x.x is the IP address of the Iperf server.

Note: You must have IP connectivity between the two devices, otherwise Iperf will fail.

In this demonstration, the Windows 7 laptop is the Iperf client and the OG150 is the Iperf server. The choice of client and server in my topology is a mute point, it doesn't really matter which device is client or server.....

Next, start the Iperf server on the OG150 as shown in Screenshot 5. This will use some default settings, notably TCP is being used on port 5001 (you can confirm that the OG150 is listening on TCP port 5001 by entering 'netstat -l' at the CLI). At this point, the Iperf server is simply waiting for a client to connect to it on TCP port 5001 – it will wait indefinitely. Note: To stop the Iperf server, enter 'CTRL + C'.

Screenshot 5 – Start the Iperf server on the OG150

```
root@OG150:~# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 kByte (default)
-----
█
```

Next, on the Iperf client (Windows 7 laptop) try and connect to the Iperf server (OG150) as shown in Screenshot 6.



www.og150.com

Follow @theog150

Screenshot 6 – Start the Iperf client on the Windows 7 laptop

```
C:\>iperf -c 192.168.11.18
```

The Iperf client will then attempt to connect to the Iperf server. As you can see in Screenshot 7, the Iperf client (192.168.11.13) has connected to the Iperf server (192.168.11.18) on TCP port 5001 – the test is underway!

Screenshot 7 – Iperf test in progress

```
root@OG150:~# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 4] local 192.168.11.18 port 5001 connected with 192.168.11.13 port 51700
```

The Iperf test will terminate, by itself, after 10 seconds (default setting) so please be patient. Once the test has finished, the Iperf tool will display the throughput! As you can see in Screenshot 8 and Screenshot 9, both devices report the throughput/bandwidth as 94.6Mb/s – this is expected as they were both connected to a 100Mb/s Ethernet switch ☺

Screenshot 8 – OG150 Wired Iperf test result

```
root@OG150:~# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 4] local 192.168.11.18 port 5001 connected with 192.168.11.13 port 51700
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.0-10.0 sec  113 MBytes  94.6 Mbits/sec
```

Screenshot 9 – Windows 7 Laptop Wired Iperf test result

```
C:\>iperf -c 192.168.11.18
-----
Client connecting to 192.168.11.18, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[168] local 192.168.11.13 port 51700 connected with 192.168.11.18 port 5001
[ ID] Interval      Transfer    Bandwidth
[168] 0.0-10.0 sec  113 MBytes  94.6 Mbits/sec
C:\>
```

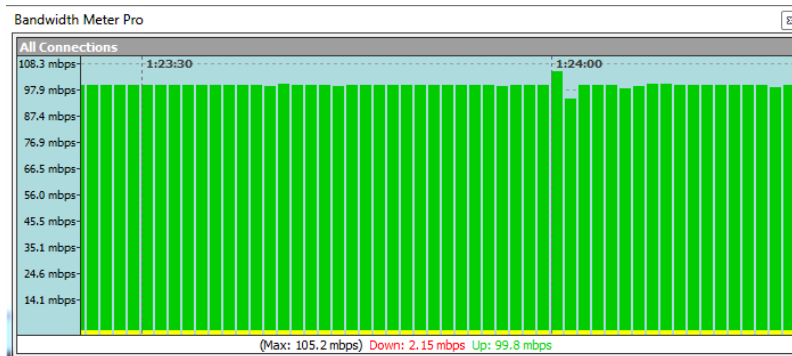
To validate the Iperf results, I use a tool on my Windows 7 laptop called 'Bandwidth Meter Pro'. This tool displays the bandwidth used by the computers network interface(s). Screenshot 10 was captured during an Iperf test, as you can see the Windows 7 laptop was transmitting (uploading) 99.8Mb/s of bandwidth. According to Screenshot 8 and Screenshot 9, 94.6Mb/s of this bandwidth was used by the Iperf test. This concludes the wired Iperf test.



www.og150.com

Follow @theog150

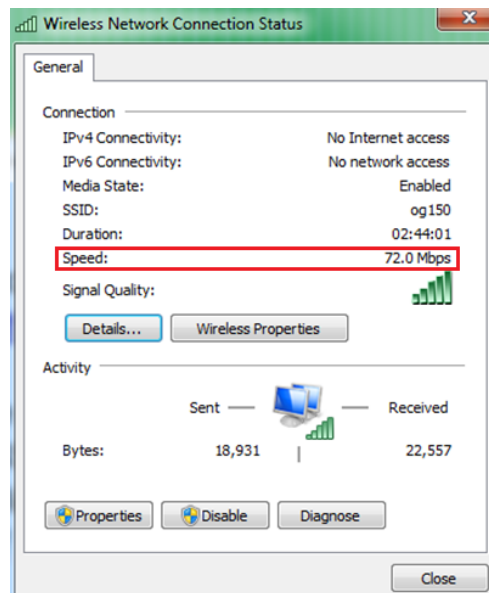
Screenshot 10 – ‘Bandwidth Meter Pro’ screenshot during wired Iperf test



Note: ‘Bandwidth Meter Pro’ currently costs \$24.95 USD and can be bought from the official website: <http://www.bandwidth-meter.net/>

I will now conduct the same test using a wireless connection. In this test the Windows 7 laptop is associated to the SSID ‘og150’ that is advertised by the OG150 (the OG150 is an access-point). Due to the close proximity of the two devices, Windows 7 reports a data rate of 72Mb/s – this is shown in Screenshot 11.

Screenshot 11 – Windows 7 laptop, 72Mb/s wireless data rate



The same commands as displayed in Screenshot 5 and Screenshot 6 were entered to start the wireless throughput test. As you can see in Screenshot 12 and Screenshot 13, both devices report the throughput/bandwidth as 38.0Mb/s. You may be thinking something is



www.og150.com

Follow @theog150

wrong here, the data rate reported in Screenshot 11 says 72Mb/s and the Iperf throughput says 38Mb/s is achieved!?!? It is very important, in fact extremely important, to understand the difference between throughput and data rates particularly when discussing wireless technologies. The *'Certified Wireless Network Administrator 4th Edition Official Study Guide'* book states:

"Defined, *data rate* is the measurement of the total amount of data that can be transferred through the system, including intentionally transferred data and overhead data. *Throughput* is defined as the amount of useful information that can be intentionally transferred through the system".

In other words, a lot of the available wireless bandwidth is consumed by wireless management and redundant data in the wireless signals. As a result, the 'achieved' throughput is much lower than the advertised data rate. The *'802.11 Wireless Networks - The Definitive Guide'* book states:

"...a practical rule of thumb is to expect 50-60% of the nominal bit rate..."

In our wireless test, the throughput was 38Mb/s and the data rate was 72Mb/s - this equates to the throughput being approximately 53% of the advertised data rate – well within the 'practical rule of thumb'.

Screenshot 12 – OG150 Wireless Iperf test result

```
root@og150:~# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 4] local 10.150.150.150 port 5001 connected with 10.150.150.94 port 53455
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.0-10.1 sec  45.5 MBytes 38.0 Mbits/sec
```

Screenshot 13 – Windows 7 Laptop Wireless Iperf test result

```
C:\>iperf -c 10.150.150.150
-----
Client connecting to 10.150.150.150, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[164] local 10.150.150.94 port 53455 connected with 10.150.150.150 port 5001
[ ID] Interval      Transfer    Bandwidth
[164] 0.0-10.0 sec  45.5 MBytes 38.0 Mbits/sec
C:\>
```

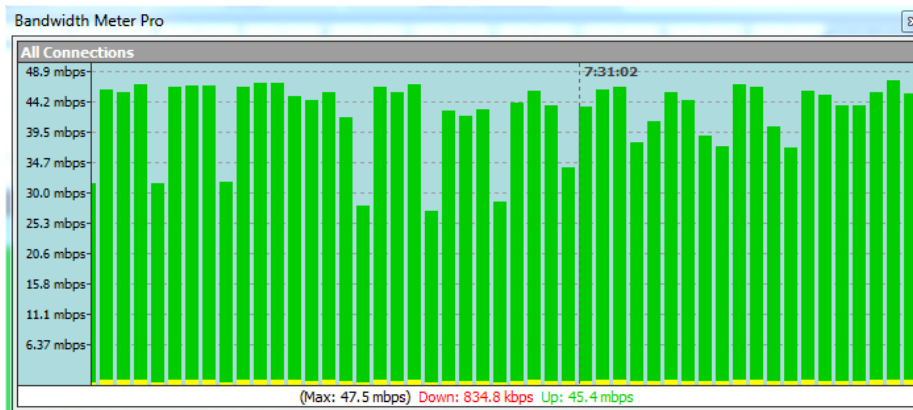
To validate the findings, I used 'Bandwidth Meter Pro' again. As you can see in Screenshot 14, the wireless throughput level fluctuates more than it does on a wired network (Screenshot 10), this is expected and could be due to many reasons (interference, someone is sending traffic on the same channel at the same time, multi-path, etc).



www.og150.com

Follow @theog150

Screenshot 14 – 'Bandwidth Meter Pro' screenshot during wireless Iperf test



Final Conclusions.

In order to wrap up this tutorial, I have drawn the following conclusions;

- Iperf is a very easy to use tool if you want to test throughput between two devices.
- It is not a replacement for commercial products that allow you to generate lots of traffic streams with specific TCP/UDP flags and QoS (Quality of Service) settings. Note: QoS is supported by Iperf but I have found this to be poorly implemented.
- Iperf can be used as part of a hackers toolkit – DOS attacks (bandwidth starvation) is one example.
- When testing wireless throughput, expect approximately 50-60% of the data rate to be realised for actual throughput.