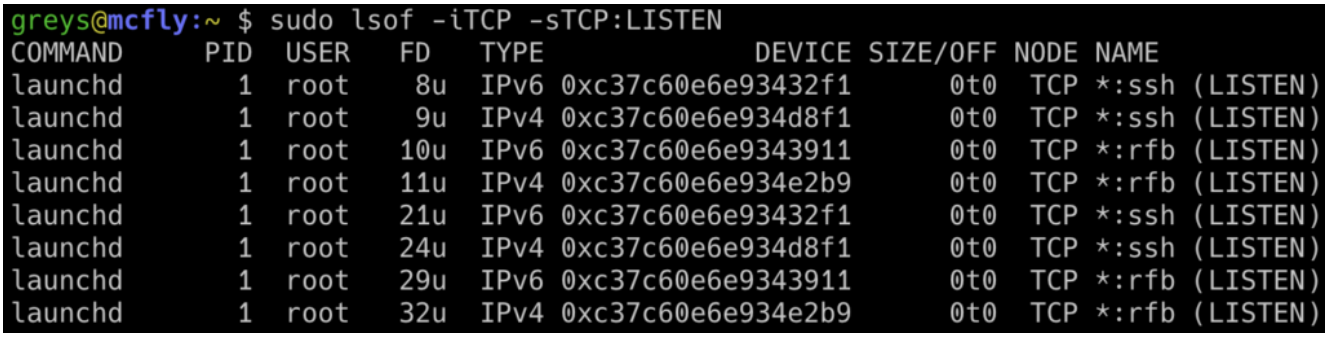
Confirm Actively Listening TCP Ports with lsof



Listening TCP ports in macOS

I was researching SSH daemon configuration on my macOS Catalina system and realised that lsof is still the best tool for meaningfully confirming network ports that are LISTENed to.

What does it mean for port to LISTEN?

All the network services in Linux (and Windows, actually) operating systems start with the same basic pattern: some process is managing incoming network connections.

Nowadays most of network services are directly managing their own network connections – meaning service like **SSH daemon** or Apache web server are made available via main process (sshd or httpd in my examples) constantly running and waiting for incoming network connections on specific port. For SSH server, default port is 22. For web servers, default ports are 80 and

443.

When we say a port is LISTENing, it means there's a process running on your system that's monitoring this specific port. SSH is therefore listening on/for port 22, Apache (httpd) is listening for port 80 and possibly 443.

Meaningful TCP ports reporting vs Default

You may remember that netstat command shows network ports as well, but its implementations are sometimes limited to just confirming that a certain port is listened to, without helping us understand what process is doing that port listening:

```
greys@mcfly:~ $ netstat -na |grep LISTEN | grep 22
tcp46       0        0   *.22000                        *.*
LISTEN
tcp4        0        0   *.22                           *.*
LISTEN
tcp6        0        0   *.22                           *.*
LISTEN
```

That's why I prefer the **lsof tool** – it's reporting processes information, which means any output you get is bound to contain processes numbers (PIDs) and most likely process names (binary names like sshd or httpd).

Use lsof to show listening TCP ports

lsof command has specific options for reporting processes with network activity: -iTCP will report TCP specific information, and -sTCP:LISTEN qualifier will filter just the processes that are listening for incoming connections on TCP ports (rather than client processes that only initiate outgoing network connections).

Don’t Forget to run lsof with sudo!

Normally **lsof** is super useful even with standard user privileges, but since I’m investigating a system service (SSH server), I have to run **lsof** as root. Otherwise **lsof** will report a list of network services, but hide the ones running above your standard user privilege level.

On modern Linux servers, **lsof** without **sudo** won’t show me anything:

```
greys@s2:~ $ lsof -iTCP -sTCP:LISTEN
greys@s2:~ $
```

My complete command to list all the services listening for incoming TCP connections in will therefore look like this (this is the macOS example):

```
greys@mcfly:~ $ sudo lsof -iTCP -sTCP:LISTEN
```

```
COMMAND     PID  USER   FD   TYPE             DEVICE SIZE/OFF
NODE NAME
launchd       1  root    8u  IPv6 0xc37c60e6e93432f1      0t0
TCP *:ssh (LISTEN)
launchd       1  root    9u  IPv4 0xc37c60e6e934d8f1      0t0
TCP *:ssh (LISTEN)
launchd       1  root   10u  IPv6 0xc37c60e6e9343911      0t0
TCP *:rfb (LISTEN)
launchd       1  root   11u  IPv4 0xc37c60e6e934e2b9      0t0
TCP *:rfb (LISTEN)
launchd       1  root   21u  IPv6 0xc37c60e6e93432f1      0t0
TCP *:ssh (LISTEN)
launchd       1  root   24u  IPv4 0xc37c60e6e934d8f1      0t0
TCP *:ssh (LISTEN)
launchd       1  root   29u  IPv6 0xc37c60e6e9343911      0t0
TCP *:rfb (LISTEN)
launchd       1  root   32u  IPv4 0xc37c60e6e934e2b9      0t0
TCP *:rfb (LISTEN)
launchd       1  root   40u  IPv6 0xc37c60e6e9342cd1      0t0
TCP localhost:intu-ec-client (LISTEN)
launchd       1  root   46u  IPv6 0xc37c60e6e9342cd1      0t0
TCP localhost:intu-ec-client (LISTEN)
launchd       1  root   47u  IPv4 0xc37c60e6e934cf29      0t0
TCP localhost:intu-ec-client (LISTEN)
launchd       1  root   48u  IPv4 0xc37c60e6e934cf29      0t0
TCP localhost:intu-ec-client (LISTEN)
kdc         120  root    5u  IPv6 0xc37c60e6e93426b1      0t0
TCP *:kerberos (LISTEN)
kdc         120  root    7u  IPv4 0xc37c60e6e934f649      0t0
TCP *:kerberos (LISTEN)
rapportd    625 greys    4u  IPv4 0xc37c60e6ef7969d9      0t0
TCP *:49263 (LISTEN)
rapportd    625 greys    5u  IPv6 0xc37c60e6e9342091      0t0
TCP *:49263 (LISTEN)
ARDAgent    697 greys    9u  IPv6 0xc37c60e6e9341a71      0t0
TCP *:net-assistant (LISTEN)
Dropbox    1148 greys  128u  IPv4 0xc37c60e6eefd8561      0t0
TCP *:17500 (LISTEN)
Dropbox    1148 greys  129u  IPv6 0xc37c60e703f137b1      0t0
TCP *:17500 (LISTEN)
Dropbox    1148 greys  154u  IPv4 0xc37c60e6f88c09d9      0t0
```

```
TCP localhost:17603 (LISTEN)
Dropbox    1148 greys  168u  IPv4 0xc37c60e6f8940011      0t0
TCP localhost:17600 (LISTEN)
dynamicli  1204 greys    7u  IPv4 0xc37c60e6f908f9d9      0t0
TCP localhost:51456 (LISTEN)
dynamicli  1205 greys    7u  IPv4 0xc37c60e6f9a453a1      0t0
TCP localhost:51549 (LISTEN)
dynamicli  1205 greys   16u  IPv4 0xc37c60e6f9b0a8f1      0t0
TCP localhost:51551 (LISTEN)
com.docke  1341 greys    7u  IPv4 0xc37c60e6fa8222b9      0t0
TCP localhost:51725 (LISTEN)
com.docke  1345 greys   15u  IPv4 0xc37c60e6eefd7b99      0t0
TCP localhost:sun-sr-https (LISTEN)
```

Confirming what process is listening on a specific port

While we're at it, here's how the previous command can be modified to confirm a specific service listening on SSH port 22:

```
greys@mcfly:~ $ sudo lsof -iTCP -sTCP:LISTEN | grep ssh
Password:
launchd       1  root    8u  IPv6 0xc37c60e6e93432f1      0t0
TCP *:ssh (LISTEN)
launchd       1  root    9u  IPv4 0xc37c60e6e934d8f1      0t0
TCP *:ssh (LISTEN)
launchd       1  root   21u  IPv6 0xc37c60e6e93432f1      0t0
TCP *:ssh (LISTEN)
launchd       1  root   24u  IPv4 0xc37c60e6e934d8f1      0t0
TCP *:ssh (LISTEN)
```

IMPORTANT: Note that because SSH is a standard service, **lsof** reports its name (ssh) rather than port number (22) in the

last column of output. **TCP *:ssh** means process is listening for TCP port for SSH service.

I was expecting **sshd**, actually. But turns out remote access via SSH is managed by launchd process in recent macOS versions. Once someone logs in though, you'll see sshd process spun up to manage the connection.

That's it for today, hope you learned something new!

See Also

- **SSH reference**
- **SSH port**
- lsof command
- Show TCP connections with lsof
- How to use lsof command