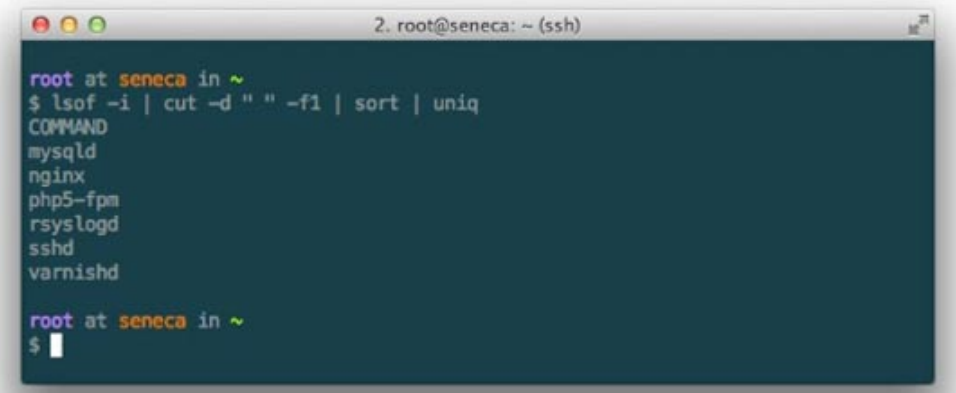
DANIELMIESSLER

An lsof Primer

By DANIEL MIESSLER (HTTPS://DANIELMIESSLER.COM/BLOG/AUTHOR/DANIEL/)

CREATED/UPDATED: SEPTEMBER 18, 2019



`lsof` is the sysadmin/security über-tool. I use it most for getting network connection related information from a system, but that's just the beginning for this powerful and too-little-known application. The tool is aptly called lsof because it "**list**s **open files**". And remember, in UNIX just about everything (including a network socket) is a file.

Interestingly, `lsof` is also the Linux/Unix command with the most switches. It has so many it has to use both minuses *and* pluses

```
usage: [-?abhlnNoOPRstUvV] [+|-c c] [+|-d s] [+D D] [+|-f[cgG]]
 [-F [f]] [-g [s]] [-i [i]] [+|-L [l]] [+|-M] [-o [o]]
 [-p s] [+|-r [t]] [-S [t]] [-T [t]] [-u s] [+|-w] [-x [fl]] [--] [names]
```

As you can see, `lsof` has a truly staggering number of options. You can use it to get information about devices on your system, what a given user is touching at any given point, or even what files or network connectivity a process is using.

For me, `lsof` replaces both `netstat` and `ps` entirely. It has everything I get from those tools and much, much more. So let's look at some of its primary capabilities:

Key Options

It's important to understand a few key things about how `lsof` works. Most importantly, when you're passing options to it, the default behavior is to OR the results. So if you are pulling a list of ports with `-i` and also a process list with `-p` you're by default going to get both results.

Here are a few others like that to keep in mind:

- **default** : without options, `lsof` lists all open files for active processes
- **grouping** : it's possible to group options, e.g. `-abC`, but you have to watch for which options take parameters
- `-a` : AND the results (instead of OR)
- `-l` : show the userID instead of the username in the output

- **-h** : get help
- **-t** : get process IDs only
- **-U** : get the UNIX socket address
- **-F** : the output is ready for another command, which can be formatted in various ways, e.g. -F pcfn (for process id, command name, file descriptor, and file name, with a null terminator)

Getting Information About the Network

As I said, one of my main usecases for `lsof` is getting information about how my system is interacting with the network. Here are some staples for getting this info:

Show all connections with `-i`

Some like to use `netstat` to get network connections, but I much prefer using `lsof` for this. The display shows things in a format that's intuitive to me, and I like knowing that from there I can simply change my syntax and get more information using the same command.

```
# lsof -i
```

```
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
dhcpcd 6061 root 4u IPv4 4510 UDP *:bootpc
sshd 7703 root 3u IPv6  6499 TCP *:ssh (LISTEN)
sshd 7892 root 3u IPv6  6757 TCP 10.10.1.5:ssh->192.168.1.5:49901 (ESTABLISHE
```

Get only IPv6 traffic with `-i 6`

```
# lsof -i 6
```

Show only TCP connections (works the same for UDP)

You can also show only TCP or UDP connections by providing the protocol right after the `-i`.

```
# lsof -iTCP
```

```
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
sshd 7703 root 3u IPv6 6499 TCP *:ssh (LISTEN)
sshd 7892 root 3u IPv6 6757 TCP 10.10.1.5:ssh->192.168.1.5:49901 (ESTABLISHED
```

Show networking related to a given port using `-i :port`

Or you can search by port instead, which is great for figuring out what's preventing another app from binding to a given port.

```
# lsof -i :22
```

```
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
sshd 7703 root 3u  IPv6 6499 TCP *:ssh (LISTEN)
sshd 7892 root 3u  IPv6 6757 TCP 10.10.1.5:ssh->192.168.1.5:49901 (ESTABLISHE
```

Show connections to a specific host using `@host`

This is quite useful when you're looking into whether you have open connections with a given host on the network or on the internet.

```
# lsof -i@172.16.12.5
```

```
sshd 7892 root 3u IPv6 6757 TCP 10.10.1.5:ssh->172.16.12.5:49901 (ESTABLISHED
```

Show connections based on the host and the port using `@host:port`

You can also combine the display of host and port.

```
# lsof -i@172.16.12.5:22
```

```
sshd 7892 root 3u IPv6 6757 TCP 10.10.1.5:ssh->192.168.1.5:49901 (ESTABLISHED
```

Find listening ports

Find ports that are awaiting connections.

```
# lsof -i -sTCP:LISTEN
```

You can also do this by grepping for "LISTEN" as well.

```
# lsof -i | grep -i LISTEN
```

```
iTunes    400 daniel   16u  IPv4 0x4575228  0t0 TCP *:daap (LISTEN)
```

Find established connections

You can also show any connections that are already pinned up.

```
# lsof -i -sTCP:ESTABLISHED
```

You can also do this just by searching for "ESTABLISHED" in the output via `grep`.

```
# lsof -i | grep -i ESTABLISHED
```

```
firefox-b 169 daniel  49u IPv4 0t0 TCP 1.2.3.3:1863->1.2.3.4:http (ESTABLISHED)
```

User Information

You can also get information on various users and what they're doing on the system, including their activity on the network, their interactions with files, etc.

Show what a given user has open using `-u`

```
# lsof -u daniel
```

```
-- snipped --
Dock 155 daniel  txt REG   14,2   2798436   823208 /usr/lib/libicucore.A.dylib
Dock 155 daniel  txt REG   14,2   1580212   823126 /usr/lib/libobjc.A.dylib
Dock 155 daniel  txt REG   14,2   2934184   823498 /usr/lib/libstdc++.6.0.4.dylib
Dock 155 daniel  txt REG   14,2    132008   823505 /usr/lib/libgcc_s.1.dylib
Dock 155 daniel  txt REG   14,2    212160   823214 /usr/lib/libauto.dylib
-- snipped --
```

Show what all users are doing except a certain user using `-u ^user`

```
# lsof -u ^daniel
```

```
-- snipped --
Dock 155 jim  txt REG   14,2   2798436   823208 /usr/lib/libicucore.A.dylib
Dock 155 jim  txt REG   14,2   1580212   823126 /usr/lib/libobjc.A.dylib
Dock 155 jim  txt REG   14,2   2934184   823498 /usr/lib/libstdc++.6.0.4.dylib
Dock 155 jim  txt REG   14,2    132008   823505 /usr/lib/libgcc_s.1.dylib
Dock 155 jim  txt REG   14,2    212160   823214 /usr/lib/libauto.dylib
-- snipped --
```

Kill everything a given user is doing

It's nice to be able to nuke everything being run by a given user.

```
# kill -9 `lsof -t -u daniel`
```

Commands and Processes

It's often useful to be able to see what a given program or process is up to, and with `lsof` you can do this by name or by process ID. Here are a few options:

See what files and network connections a named command is using with `-c`

```
# lsof -c syslog-ng
```

```
COMMAND    PID USER   FD   TYPE     DEVICE    SIZE      NODE NAME
syslog-ng 7547 root  cwd    DIR    3,3    4096   2 /
syslog-ng 7547 root  rtd    DIR    3,3    4096   2 /
syslog-ng 7547 root  txt    REG    3,3  113524  1064970 /usr/sbin/syslog-ng
-- snipped --
```

See what a given process ID has open using `-p`

```
# lsof -p 10075
```

```
-- snipped --
sshd    10068 root  mem    REG    3,3   34808 850407 /lib/libnss_files-2.4.so
sshd    10068 root  mem    REG    3,3   34924 850409 /lib/libnss_nis-2.4.so
sshd    10068 root  mem    REG    3,3   26596 850405 /lib/libnss_compat-2.4.so
sshd    10068 root  mem    REG    3,3  200152 509940 /usr/lib/libssl.so.0.9.7
sshd    10068 root  mem    REG    3,3   46216 510014 /usr/lib/liblber-2.3
sshd    10068 root  mem    REG    3,3   59868 850413 /lib/libresolv-2.4.so
sshd    10068 root  mem    REG    3,3 1197180 850396 /lib/libc-2.4.so
sshd    10068 root  mem    REG    3,3   22168 850398 /lib/libcrypt-2.4.so
sshd    10068 root  mem    REG    3,3   72784 850404 /lib/libnsl-2.4.so
sshd    10068 root  mem    REG    3,3   70632 850417 /lib/libz.so.1.2.3
sshd    10068 root  mem    REG    3,3    9992 850416 /lib/libutil-2.4.so
-- snipped --
```

The `-t` option returns just a PID

```
# lsof -t -c Mail
```

```
350
```

Files and Directories

By looking at a given file or directory you can see what all on the system is interacting with it–including users, processes, etc.

Show everything interacting with a given directory

```
# lsof /var/log/messages/
```

```
COMMAND    PID USER   FD   TYPE DEVICE   SIZE   NODE NAME
syslog-ng 7547 root    4w   REG    3,3 217309 834024 /var/log/messages
```

Show everything interacting with a given file

```
# lsof /home/daniel/firewall_whitelist.txt
```

Advanced Usage

Similar to `tcpdump` (https://danielmiessler.com/study/tcpdump/), the power really shows itself when you start combining queries.

Show me everything daniel is doing connected to 1.1.1.1

```
# lsof -u daniel -i @1.1.1.1
```

```
bkdr   1893 daniel 3u  IPv6 3456 TCP 10.10.1.10:1234->1.1.1.1:31337 (ESTABLISH
```

Using the `-t` and `-c` options together to HUP processes

```
# kill -HUP `lsof -t -c sshd`
```

Show open connections with a port range

```
# lsof -i @fw.google.com:2150-2180
```

Conclusion

This primer just scratches the surface of `lsof`'s functionality. For a full reference, run `man lsof` or check out the online version (https://www.netadmintools.com/html/lsof.man.html). I hope this has been useful to you, and as always, comments and corrections are welcomed.

Notes

1. The `lsof` man page:
   http://www.netadmintools.com/html/lsof.man.html
   (https://www.netadmintools.com/html/lsof.man.html)

CREATED: FEBRUARY 2009 | *UPDATED: DECEMBER 2015*