Linux Lab 9 Unnecessary services

An important part of securing a computer is making sure that only necessary daemons/services (I tend to use the two terms interchangeably) are running. Anything that you don't need presents an unnecessary security risk, especially if listens on the network. So, it's important to be able to find and remove unnecessary services. You do have to be careful, however. You will break things if you turn off daemons that are more important than you thought…do this in a test environment first!

Locate listening network services

Note: netstat vs. ss

The venerable and useful application `netstat` has been deprecated and replaced by `ss`. It appears that netstat was not being maintained. You can still run `netstat` if you install the net-tools package.

```
sudo apt-get install net-tools
```

The usage and output for netstat and ss are similar. We'll show both.

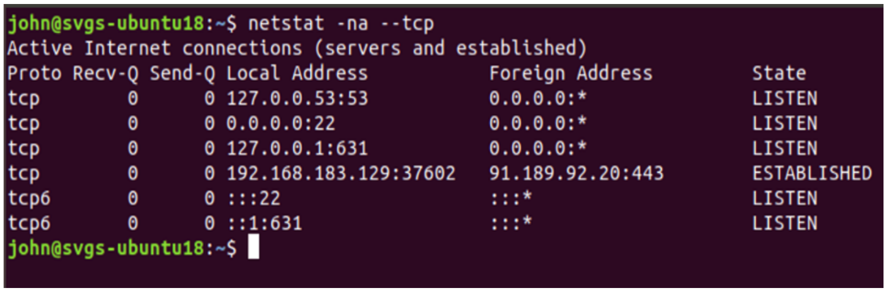
netstat

Unneeded services that listen to the network are potentially dangerous. If they are poorly configured or out of date, they may make the computer vulnerable to attack. Use the commands,

```
netstat -na --tcp
netstat -na --udp
```

to locate listening ports. The ports that are listening, with a local address of 0.0.0.0, are the ones that allow connections from the outside. When the local address is 127.0.0.1, the computer is listening for connections from itself (inter process communication.) Record the listening ports. In the example below, look at the Local Address column. The first and third lines are listening on the internal loopback address 127.x.x.x which is only accessible from the local computer; they are used for inter-process communication. Line 2, the Local Address is 0.0.0.0, which means any interface on the computer. Line 2 means external hosts can connect to this computer on port 22 (SSH). Line 4 shows that this computer's interface on 192.168.183.129, port 37602, is connected to a server at 91.189.92.20 on port 443 (HTTPS.) Line 2, the one that says the computer is listening for external connections, is most important.

127.0.0.53:53 internal listen on port 53
0.0.0.0:22 listen for EXTERNAL connection on port 22
127.0.0.1:631 internal listen on port 631
192.168.183.129:37602 external connection to
91.189.92.20 port 443 HTTPS in process



```
john@svgs-ubuntu18:~$ netstat -na --tcp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 192.168.183.129:37602   91.189.92.20:443        ESTABLISHED
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
john@svgs-ubuntu18:~$ 
```

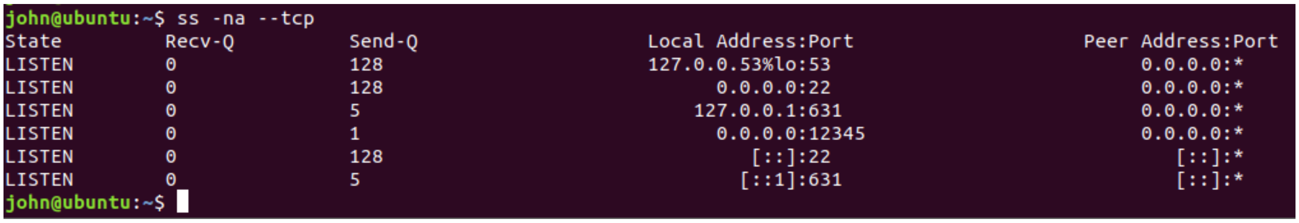
Note: If you use netstat without the --tcp or --udp option you will see Unix STREAM connections. These are internal connections, and there are a lot of them. If you're only interested in external network connections, they clutter the output.

ss

The commands for `ss` are

```
ss -na --tcp
ss -na --udp
```

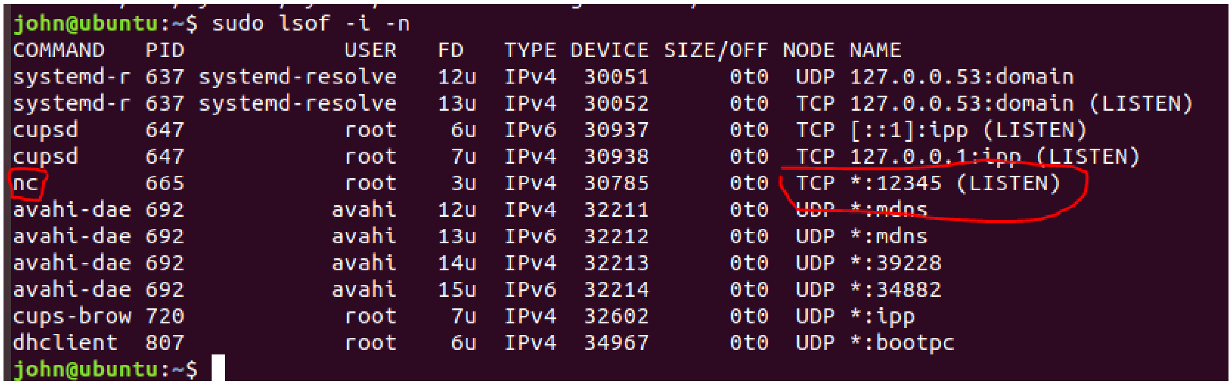
(However, the `–t` and `–u` options work as well as `--tcp` and `--udp`. I used the longer options because they are the same as `netstat`.)



```
john@ubuntu:~$ ss -na --tcp
State          Recv-Q          Send-Q                 Local Address:Port                  Peer Address:Port
LISTEN         0               128                  127.0.0.53%lo:53                          0.0.0.0:*
LISTEN         0               128                        0.0.0.0:22                          0.0.0.0:*
LISTEN         0               5                        127.0.0.1:631                         0.0.0.0:*
LISTEN         0               1                          0.0.0.0:12345                       0.0.0.0:*
LISTEN         0               128                           [::]:22                             [::]:*
LISTEN         0               5                            [::1]:631                            [::]:*
john@ubuntu:~$
```

lsof

Another command that will help is lsof (list open files). With the -i option, lsof lists files that have open IP connections. The data in the NAME column in the lsof output begins with *: and contains (LISTEN), for services that are listening for outside connections. Inside connections will have 127.0.0.1 (IP version 4) or [::1] (IP version 6) instead of *. In the example below, you will see that I have nc (netcat) listening on port 12345; not good. I was playing with a netcat backdoor and forgot to turn it off. Oops.

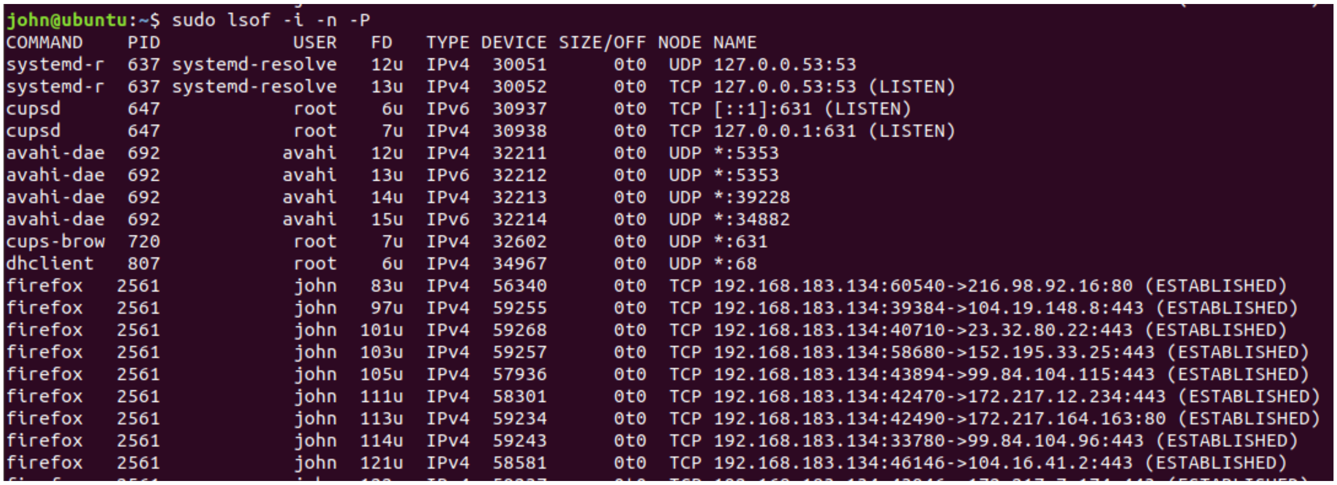


```
john@ubuntu:~$ sudo lsof -i -n
COMMAND    PID            USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
systemd-r 637 systemd-resolve   12u  IPv4  30051      0t0  UDP 127.0.0.53:domain
systemd-r 637 systemd-resolve   13u  IPv4  30052      0t0  TCP 127.0.0.53:domain (LISTEN)
cupsd     647            root    6u  IPv6  30937      0t0  TCP [::1]:ipp (LISTEN)
cupsd     647            root    7u  IPv4  30938      0t0  TCP 127.0.0.1:ipp (LISTEN)
nc        665            root    3u  IPv4  30785      0t0  TCP *:12345 (LISTEN)
avahi-dae 692           avahi   12u  IPv4  32211      0t0  UDP *:mdns
avahi-dae 692           avahi   13u  IPv6  32212      0t0  UDP *:mdns
avahi-dae 692           avahi   14u  IPv4  32213      0t0  UDP *:39228
avahi-dae 692           avahi   15u  IPv6  32214      0t0  UDP *:34882
cups-brow 720            root    7u  IPv4  32602      0t0  UDP *:ipp
dhclient  807            root    6u  IPv4  34967      0t0  UDP *:bootpc
john@ubuntu:~$
```

Use lsof with and without the -P option, so you can see both the port name and port number. Note: Be sure to run lsof with root privileges.

```
sudo lsof -i -n
sudo lsof -i -n -P
```

This shows lsof output when Firefox has connected to the Nasa web site (ESTABLISHED connections).



```
john@ubuntu:~$ sudo lsof -i -n -P
COMMAND     PID            USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
systemd-r   637 systemd-resolve   12u  IPv4  30051      0t0  UDP 127.0.0.53:53
systemd-r   637 systemd-resolve   13u  IPv4  30052      0t0  TCP 127.0.0.53:53 (LISTEN)
cupsd       647            root    6u  IPv6  30937      0t0  TCP [::1]:631 (LISTEN)
cupsd       647            root    7u  IPv4  30938      0t0  TCP 127.0.0.1:631 (LISTEN)
avahi-dae   692           avahi   12u  IPv4  32211      0t0  UDP *:5353
avahi-dae   692           avahi   13u  IPv6  32212      0t0  UDP *:5353
avahi-dae   692           avahi   14u  IPv4  32213      0t0  UDP *:39228
avahi-dae   692           avahi   15u  IPv6  32214      0t0  UDP *:34882
cups-brow   720            root    7u  IPv4  32602      0t0  UDP *:631
dhclient    807            root    6u  IPv4  34967      0t0  UDP *:68
firefox    2561            john   83u  IPv4  56340      0t0  TCP 192.168.183.134:60540->216.98.92.16:80 (ESTABLISHED)
firefox    2561            john   97u  IPv4  59255      0t0  TCP 192.168.183.134:39384->104.19.148.8:443 (ESTABLISHED)
firefox    2561            john  101u  IPv4  59268      0t0  TCP 192.168.183.134:40710->23.32.80.22:443 (ESTABLISHED)
firefox    2561            john  103u  IPv4  59257      0t0  TCP 192.168.183.134:58680->152.195.33.25:443 (ESTABLISHED)
firefox    2561            john  105u  IPv4  57936      0t0  TCP 192.168.183.134:43894->99.84.104.115:443 (ESTABLISHED)
firefox    2561            john  111u  IPv4  58301      0t0  TCP 192.168.183.134:42470->172.217.12.234:443 (ESTABLISHED)
firefox    2561            john  113u  IPv4  59234      0t0  TCP 192.168.183.134:42490->172.217.164.163:80 (ESTABLISHED)
firefox    2561            john  114u  IPv4  59243      0t0  TCP 192.168.183.134:33780->99.84.104.96:443 (ESTABLISHED)
firefox    2561            john  121u  IPv4  58581      0t0  TCP 192.168.183.134:46146->104.16.41.2:443 (ESTABLISHED)
```

Run `lsof` and record the listening ports, port names, and command (service names) you see.

(This command only applies to Ubuntu 15 and higher--skip this if you are using Ubuntu 14)
We can see which services opened sockets (network connections) through systemd by using the command,
`systemctl list-units --type socket`
The list of services may not be identical to the list of ports from the netstat command, if services opened sockets outside of systemd.

(We haven't covered nmap yet, but you have installed it on your host computer. Zenmap may be helpful in CyberPatriots.)
A final way to locate or confirm listening ports is to scan your VM from another computer. Find the IP address of your VM by executing either ifconfig (interface configuration, different from Windows ipconfig) or the newer command, ip address. Then ping your VM from your Windows host machine. Once you've verified connectivity, run a scan of your VM using nmap (or Zenmap) from your Windows host. Note: It is possible for your VM to be listening on a port, but the VM's firewall is configured to block it. Also, by default nmap only scans the 1000 most popular ports. If you have time, you can scan all 65535 ports by adding -p 0-65535 to your nmap command and running it again.

With the data you have, and assistance from your favorite search engine, determine what the listening services are doing, and whether you should shut them down. A search for "shut down xyz service" may be helpful, as there are usually questions asking what happens when the service is shut down.

Hand in

What listening services did you find, what do they do, and should you shut them down?

Other unnecessary services

This section works on Linux that is based on systemd, instead of upstart or SysV. Ubuntu 15 and later uses systemd.

The command to list all services running under systemd is,
`systemctl list-units --type service`

This puts the output into less. If you want to make a list that contains only the service names, you can use

```
systemctl list-units --type service --full | cut -f1 -d' '
```

The option, --full, causes systemctl to output results in a format that cut can read. After experimenting, I found that the delimiter is space, instead of the default tab.

The list of services is long, and it is difficult for a person new to an operating system to determine which services are necessary, and which are not. You could research each service to determine which services you need. You could also consult a security benchmark (https://www.cisecurity.org/cis-benchmarks/ for example) to create a final list. Once you are familiar with and operating system, you can create a baseline installation and keep a copy of that installation. Then as things change, you can compare your OS to the baseline installation you've changed and locate unneeded (or attacker's) service quicker.

Shut down unnecessary services

Once you have decided to shut down services, we need to do two things: shut them down and prevent them from starting when the computer reboots. The commands are,

```
systemctl stop [service name]
systemctl disable [service name]
```

Be careful here. You may find listening ports that are listed as systemd, but actually run as another service under systemd. Don't shut down systemd.