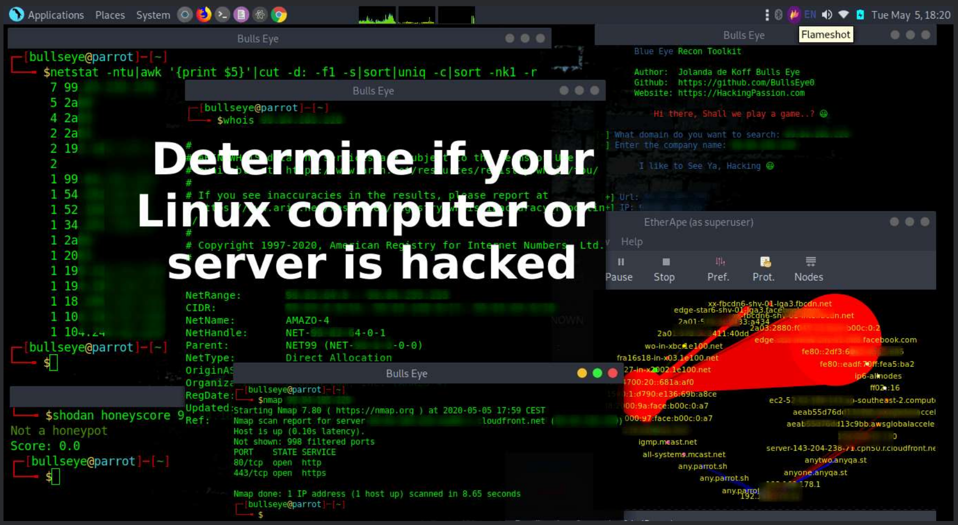
Determine if Your Linux Computer or Server Is Hacked

Bulls Eye included in linux windows
2020-05-06 2569 words 13 minutes



```
┌─[bullseye@parrot]─[~]
└──╼ $netstat -ntu|awk '{print $5}'|cut -d: -f1 -s|sort|uniq -c|sort -nk1 -r
      7 99[redacted]
      5 2a[redacted]
      4 2a[redacted]
      2 2a[redacted]
      2 19[redacted]
      2
      1 99[redacted]
      1 54[redacted]
      1 52[redacted]
      1 34[redacted]
      1 2a[redacted]
      1 20[redacted]
      1 19[redacted]
      1 19[redacted]
      1 18[redacted]
      1 10[redacted]
      1 104.24
┌─[bullseye@parrot]─[~]
└──╼ $
```

```
┌─[bullseye@parrot]─[~]
└──╼ $whois [redacted]

# If you see inaccuracies in the results, please report at

# Copyright 1997-2020, American Registry for Internet Numbers, Ltd.

NetRange:       [redacted]
CIDR:           [redacted]
NetName:        AMAZO-4
NetHandle:      NET-[redacted]4-0-1
Parent:         NET99 (NET-[redacted]-0-0)
NetType:        Direct Allocation
```

```
└──╼ $shodan honeyscore 9
Not a honeypot
Score: 0.0
┌─[bullseye@parrot]─[~]
└──╼ $
```

```
┌─[bullseye@parrot]─[~]
└──╼ $nmap [redacted]
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-05 17:59 CEST
Nmap scan report for server[redacted]loudfront.net ([redacted])
Host is up (0.10s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 8.65 seconds
┌─[bullseye@parrot]─[~]
└──╼ $
```

```
       Blue Eye Recon Toolkit

       Author:  Jolanda de Koff Bulls Eye
       Github:  https://github.com/BullsEye0
       Website: https://HackingPassion.com

          Hi there, Shall we play a game..? 😃

] What domain do you want to search: [redacted]
] Enter the company name: [redacted]

       I like to See Ya, Hacking 😃

+] Url: [redacted]
+] IP: [redacted]
```

"**Advice from a Hacker**" How do you determine whether your computer or server has been hacked. If you suspected this, this article certainly applies to you. But I also discuss several great commands that every Linux user or "Hacker" should know. And I show you various options. Some of these commands will also work for a Mac and Windows, so it's a good idea to take a look if only for the tips you might want to use.

Keep calm and don't panic if you have been

Just classify everything. Do not access a file with cat or strings, catalog the files and save that for later. Once you start removing things, you can no longer investigate how deeply they have penetrated. Don't be misled and just stay calm. Just do some investigation and research.

Take a good look at the attacker, you may find an IP address or a trace that has been left behind. This can only make the research more fun. Try to find out as much as possible about the attacker. If you have all the data then you can look to delete it safely.

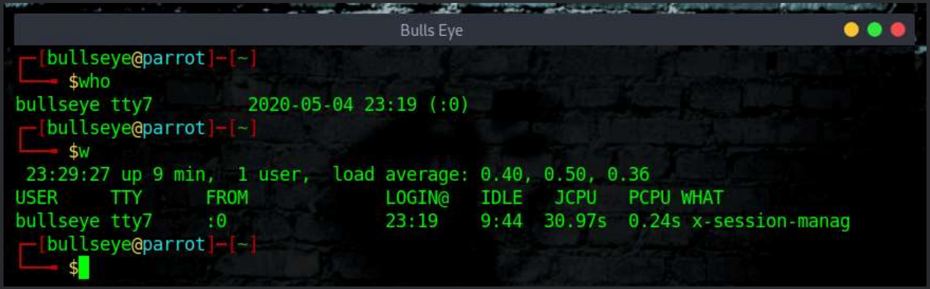
Obviously, it is urgent if you are very duped and a lot of money is involved, but then hire a team of specialized Ethical Hackers or Cyber security Experts. If you have a business that is always the best you can do.

> "What hackers do is figure out technology and experiment with it in ways many people never imagined. They also have a strong desire to share this information with others and to explain it to people whose only qualification may be the desire to learn."

Show a listing of last logged in users

w or who

The first thing you should look for is who is currently logged into your computer. It is not uncommon to find the attacker actually logged into the server and working on it.



```
┌─[bullseye@parrot]─[~]
└──╼ $who
bullseye tty7         2020-05-04 23:19 (:0)
┌─[bullseye@parrot]─[~]
└──╼ $w
 23:29:27 up 9 min,  1 user,  load average: 0.40, 0.50, 0.36
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
bullseye tty7     :0               23:19    9:44  30.97s  0.24s x-session-manag
┌─[bullseye@parrot]─[~]
└──╼ $
```

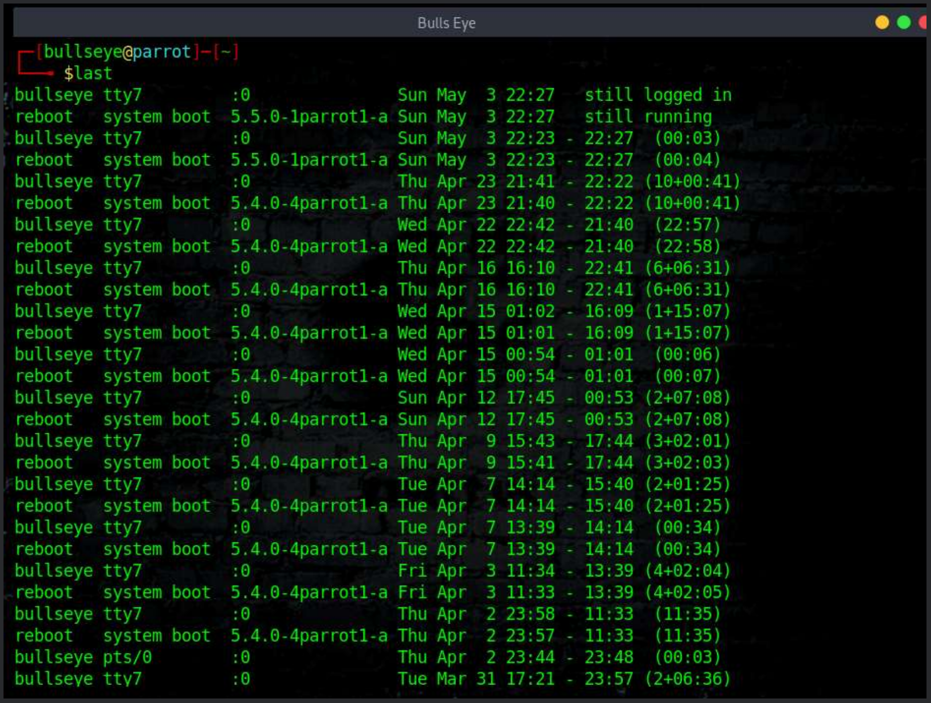
Use the command last

Show a listing of **last** logged in users. The history with this command goes all the way back to the start of the setup of the computer or server. (You can also

```
last -h
```

Code

```
Usage:
 last [options] [<username>...] [<tty>...]

Show a listing of last logged in users.

Options:
 -<number>            how many lines to show
 -a, --hostlast       display hostnames in the last column
 -d, --dns            translate the IP number back into a hostname
 -f, --file <file>    use a specific file instead of /var/log/wtmp
 -F, --fulltimes      print full login and logout times and dates
 -i, --ip             display IP numbers in numbers-and-dots notation
 -n, --limit <number> how many lines to show
 -R, --nohostname     don't display the hostname field
 -s, --since <time>   display the lines since the specified time
 -t, --until <time>   display the lines until the specified time
 -p, --present <time> display who were present at the specified time
 -w, --fullnames      display full user and domain names
 -x, --system         display system shutdown entries and run level c
     --time-format <format>  show timestamps in the specified <format
                               notime|short|full|iso

 -h, --help           display this help
 -V, --version        display version
```



```
┌─[bullseye@parrot]─[~]
└──╼ $last
bullseye tty7         :0               Sun May  3 22:27   still logged in
reboot   system boot  5.5.0-1parrot1-a Sun May  3 22:27   still running
bullseye tty7         :0               Sun May  3 22:23 - 22:27  (00:03)
reboot   system boot  5.5.0-1parrot1-a Sun May  3 22:23 - 22:27  (00:04)
bullseye tty7         :0               Thu Apr 23 21:41 - 22:22 (10+00:41)
reboot   system boot  5.4.0-4parrot1-a Thu Apr 23 21:40 - 22:22 (10+00:41)
bullseye tty7         :0               Wed Apr 22 22:42 - 21:40  (22:57)
reboot   system boot  5.4.0-4parrot1-a Wed Apr 22 22:42 - 21:40  (22:58)
bullseye tty7         :0               Thu Apr 16 16:10 - 22:41 (6+06:31)
reboot   system boot  5.4.0-4parrot1-a Thu Apr 16 16:10 - 22:41 (6+06:31)
bullseye tty7         :0               Wed Apr 15 01:02 - 16:09 (1+15:07)
reboot   system boot  5.4.0-4parrot1-a Wed Apr 15 01:01 - 16:09 (1+15:07)
bullseye tty7         :0               Wed Apr 15 00:54 - 01:01  (00:06)
reboot   system boot  5.4.0-4parrot1-a Wed Apr 15 00:54 - 01:01  (00:07)
bullseye tty7         :0               Sun Apr 12 17:45 - 00:53 (2+07:08)
reboot   system boot  5.4.0-4parrot1-a Sun Apr 12 17:45 - 00:53 (2+07:08)
bullseye tty7         :0               Thu Apr  9 15:43 - 17:44 (3+02:01)
reboot   system boot  5.4.0-4parrot1-a Thu Apr  9 15:41 - 17:44 (3+02:03)
bullseye tty7         :0               Tue Apr  7 14:14 - 15:40 (2+01:25)
reboot   system boot  5.4.0-4parrot1-a Tue Apr  7 14:14 - 15:40 (2+01:25)
bullseye tty7         :0               Tue Apr  7 13:39 - 14:14  (00:34)
reboot   system boot  5.4.0-4parrot1-a Tue Apr  7 13:39 - 14:14  (00:34)
bullseye tty7         :0               Fri Apr  3 11:34 - 13:39 (4+02:04)
reboot   system boot  5.4.0-4parrot1-a Fri Apr  3 11:33 - 13:39 (4+02:05)
bullseye tty7         :0               Thu Apr  2 23:58 - 11:33  (11:35)
reboot   system boot  5.4.0-4parrot1-a Thu Apr  2 23:57 - 11:33  (11:35)
bullseye pts/0        :0               Thu Apr  2 23:44 - 23:48  (00:03)
bullseye tty7         :0               Tue Mar 31 17:21 - 23:57 (2+06:36)
```

Code

```
tail -n 200 ~/.bash_history | more
```

Code

```
cat ~/.bash_history | more
```

Of course, you can also open an editor (Like vim or nano) and save the output. So that you can notice any changes at a **later time**. Check also command from other users that you might have on your computer. /home/**username/**

Code

```
sudo vim /home/USER_YOU_WANT_TO_VIEW/.bash_history
```

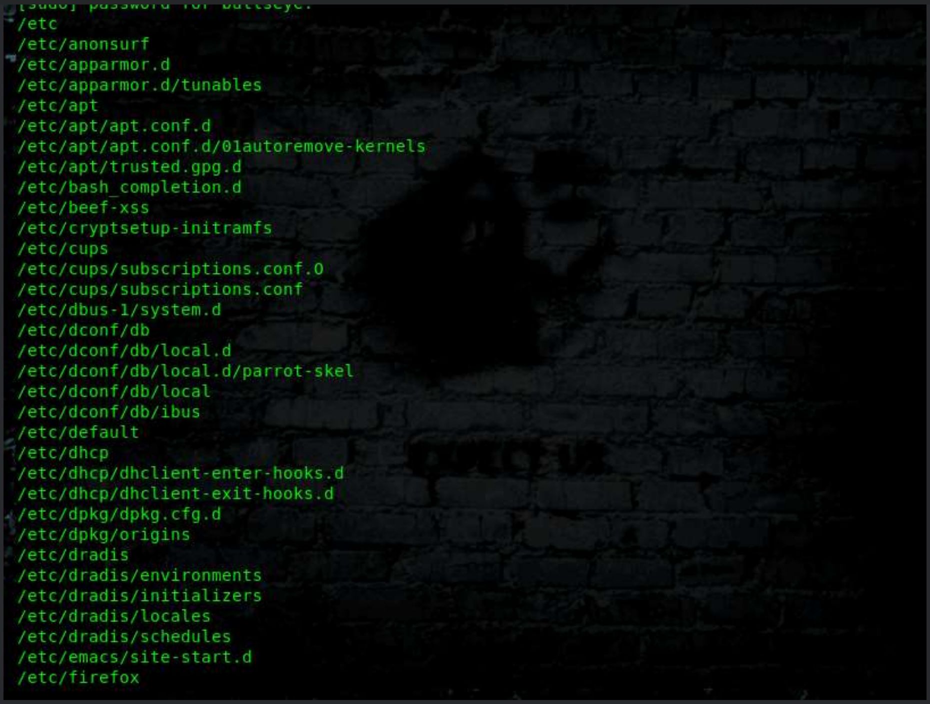
System files that have changed recently.

With this command, you can see what has happened recently. The "**-2**" means 2 days, i.e. this shows me all files modified in the last 2 days.

Code

```
sudo find /etc /var -mtime -2
```

Now if you haven't installed any new software on your server for a while then this command will run and produce very little output. Here in this picture I just did a new upgrade, so there is a lot to see.



```
[sudo] password for bullseye:
/etc
/etc/anonsurf
/etc/apparmor.d
/etc/apparmor.d/tunables
/etc/apt
/etc/apt/apt.conf.d
/etc/apt/apt.conf.d/01autoremove-kernels
/etc/apt/trusted.gpg.d
/etc/bash_completion.d
/etc/beef-xss
/etc/cryptsetup-initramfs
/etc/cups
/etc/cups/subscriptions.conf.O
/etc/cups/subscriptions.conf
/etc/dbus-1/system.d
/etc/dconf/db
/etc/dconf/db/local.d
/etc/dconf/db/local.d/parrot-skel
/etc/dconf/db/local
/etc/dconf/db/ibus
/etc/default
/etc/dhcp
/etc/dhcp/dhclient-enter-hooks.d
/etc/dhcp/dhclient-exit-hooks.d
/etc/dpkg/dpkg.cfg.d
/etc/dpkg/origins
/etc/dradis
/etc/dradis/environments
/etc/dradis/initializers
/etc/dradis/locales
/etc/dradis/schedules
/etc/emacs/site-start.d
/etc/firefox
```

Verify the current connections from your computer and or server

Netstat

Code

```
netstat --help
```

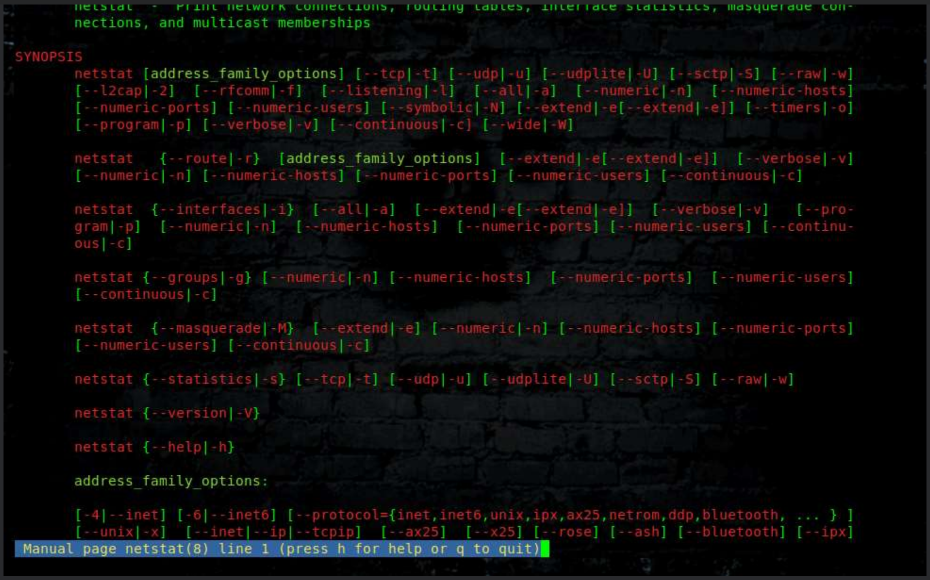
Code

```
     netstat { [-WeenNac] -i | [-cnNe] -M | -s [-6tuw] }

      -r, --route                display routing table
      -i, --interfaces           display interface table
      -g, --groups               display multicast group memberships
      -s, --statistics           display networking statistics (like
      -M, --masquerade           display masqueraded connections

      -v, --verbose              be verbose
      -W, --wide                 don't truncate IP addresses
      -n, --numeric              don't resolve names
      --numeric-hosts            don't resolve host names
      --numeric-ports            don't resolve port names
      --numeric-users            don't resolve user names
      -N, --symbolic             resolve hardware names
      -e, --extend               display other/more information
      -p, --programs             display PID/Program name for sockets
      -o, --timers               display timers
      -c, --continuous           continuous listing

      -l, --listening            display listening server sockets
      -a, --all                  display all sockets (default: connec
      -F, --fib                  display Forwarding Information Base
      -C, --cache                display routing cache instead of FIB
      -Z, --context              display SELinux security context for

<Socket>={-t|--tcp} {-u|--udp} {-U|--udplite} {-S|--sctp} {-w|--raw
         {-x|--unix} --ax25 --ipx --netrom
<AF>=Use '-6|-4' or '-A <af>' or '--<af>'; default: inet
List of possible address families (which support routing):
  inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
  netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
  x25 (CCITT X.25)
```

If you want more information about netstat, you can use the man (manual) page

Code

```
man netstat
```



```
       netstat  -  Print network connections, routing tables, interface statistics, masquerade con-
       nections, and multicast memberships

SYNOPSIS
       netstat [address_family_options] [--tcp|-t] [--udp|-u] [--udplite|-U] [--sctp|-S] [--raw|-w]
       [--l2cap|-2]  [--rfcomm|-f]  [--listening|-l]  [--all|-a]  [--numeric|-n]  [--numeric-hosts]
       [--numeric-ports] [--numeric-users] [--symbolic|-N] [--extend|-e[--extend|-e]] [--timers|-o]
       [--program|-p] [--verbose|-v] [--continuous|-c] [--wide|-W]

       netstat   {--route|-r}  [address_family_options]  [--extend|-e[--extend|-e]]  [--verbose|-v]
       [--numeric|-n] [--numeric-hosts] [--numeric-ports] [--numeric-users] [--continuous|-c]

       netstat  {--interfaces|-i}  [--all|-a]  [--extend|-e[--extend|-e]]  [--verbose|-v]   [--pro-
       gram|-p]  [--numeric|-n]  [--numeric-hosts]  [--numeric-ports] [--numeric-users] [--continu-
       ous|-c]

       netstat {--groups|-g} [--numeric|-n] [--numeric-hosts]  [--numeric-ports]  [--numeric-users]
       [--continuous|-c]

       netstat  {--masquerade|-M}  [--extend|-e] [--numeric|-n] [--numeric-hosts] [--numeric-ports]
       [--numeric-users] [--continuous|-c]

       netstat {--statistics|-s} [--tcp|-t] [--udp|-u] [--udplite|-U] [--sctp|-S] [--raw|-w]

       netstat {--version|-V}

       netstat {--help|-h}

       address_family_options:

       [-4|--inet] [-6|--inet6] [--protocol={inet,inet6,unix,ipx,ax25,netrom,ddp,bluetooth, ... } ]
       [--unix|-x]  [--inet|--ip|--tcpip]  [--ax25]  [--x25] [--rose] [--ash] [--bluetooth] [--ipx]
 Manual page netstat(8) line 1 (press h for help or q to quit)
```

Often an attacker will install a program that doesn't do anything except listen on the network port for instructions. You should look for any process that is listed as in the LISTEN or ESTABLISHED status as these processes are either waiting for a connection (**LISTEN**) or have a connection open (**ESTABLISHED**). If you don't recognize these processes use "strace" or "lsof" (below an example) to try to see what they are doing.

This command will show you 2 parts, the first is "**Active Internet connections (w/o servers)**" and the second is "**Active UNIX domain sockets (w/o servers)**"

Check both carefully because if you got a malicious script running somewhere and this script is trying to sending spam mail or try to attach other servers you can easily find here.

Code

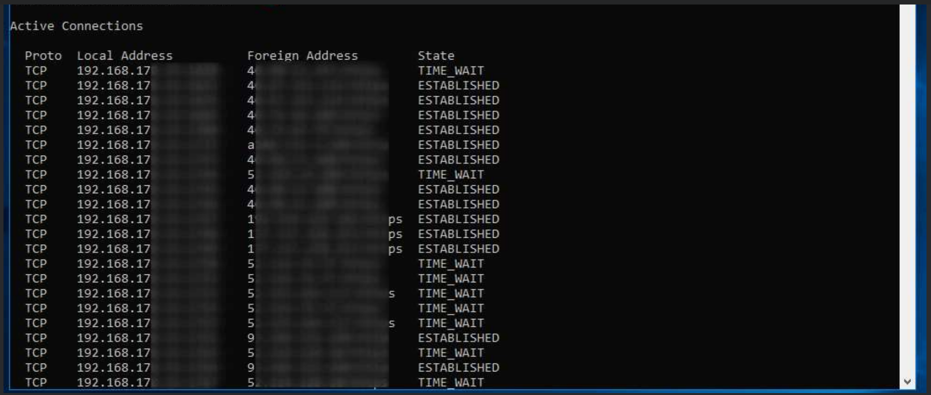
```
netstat | more
```

netstat | more Active Internet connections Determine if your Linux computer or server has been hacked

Also on **Windows** is Netstat to use. Open your Command Prompt and type:

Code

```
netstat | more
```



```
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.17[redacted]   4[redacted]            TIME_WAIT
  TCP    192.168.17[redacted]   4[redacted]            ESTABLISHED
  TCP    192.168.17[redacted]   4[redacted]            ESTABLISHED
  TCP    192.168.17[redacted]   4[redacted]            ESTABLISHED
  TCP    192.168.17[redacted]   4[redacted]            ESTABLISHED
  TCP    192.168.17[redacted]   a[redacted]            ESTABLISHED
  TCP    192.168.17[redacted]   4[redacted]            ESTABLISHED
  TCP    192.168.17[redacted]   5[redacted]            TIME_WAIT
  TCP    192.168.17[redacted]   4[redacted]            ESTABLISHED
  TCP    192.168.17[redacted]   4[redacted]            ESTABLISHED
  TCP    192.168.17[redacted]   1[redacted]            ESTABLISHED
  TCP    192.168.17[redacted]   1[redacted]            ESTABLISHED
  TCP    192.168.17[redacted]   1[redacted]            ESTABLISHED
  TCP    192.168.17[redacted]   5[redacted]            TIME_WAIT
  TCP    192.168.17[redacted]   5[redacted]            TIME_WAIT
  TCP    192.168.17[redacted]   5[redacted]            TIME_WAIT
  TCP    192.168.17[redacted]   5[redacted]            TIME_WAIT
  TCP    192.168.17[redacted]   5[redacted]            TIME_WAIT
  TCP    192.168.17[redacted]   9[redacted]            ESTABLISHED
  TCP    192.168.17[redacted]   5[redacted]            TIME_WAIT
  TCP    192.168.17[redacted]   9[redacted]            ESTABLISHED
  TCP    192.168.17[redacted]   5[redacted]            TIME_WAIT
```

Below I show an example of how to use the command **sudo netstat -atnp | grep ESTA** used. The first image without having anything open, the second image when I opened about 15 tabs in Chrome.

I must say that these commands have always been useful in the past, for example when you spoke to someone on **Telegram**, you could see the IP addresses of the people you spoke to. (also from bots). This IP address leak is now closed.

Code

```
sudo netstat -atnp | grep ESTA
```

sudo netstat -atnp | grep ESTA

When entered correctly, this command will return a descending list of which IPs are connected to your (**server**) "I use this command often for my computer" and how many connections each one has. Looking at your results, you will see connections listed ranging anywhere from 1 to about 50 connections per IP. This can be quite common for normal traffic (server). If however, you see some IPs with 100+ connections, this is something to scrutinize.

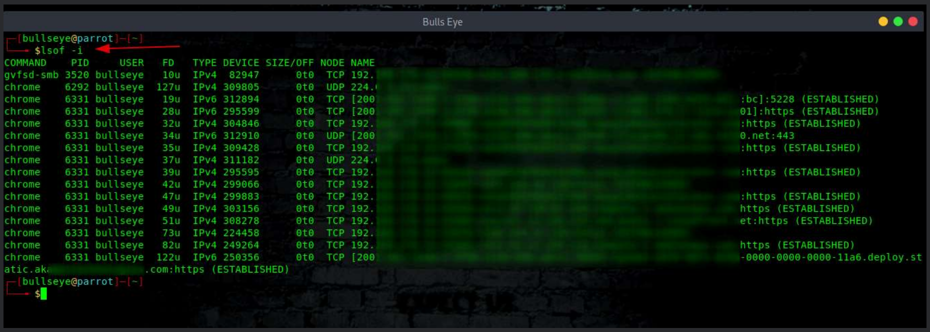
Included in the list, you may see known IPs, one or more of the server's own IPs, or even your own personal IP with many connections.

Code

```
netstat -ntu|awk '{print $5}'|cut -d: -f1 -s|sort|uniq -c|sort -nk1 -
```

netstat -ntu|awk

lsof



Bulls Eye

```
┌─[bullseye@parrot]─[~]
└──╼ $lsof -i
COMMAND     PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
gvfsd-smb 3520 bullseye   10u  IPv4  82947      0t0  TCP 192.[redacted]
chrome    6292 bullseye  127u  IPv4 309805      0t0  UDP 224.[redacted]
chrome    6331 bullseye   19u  IPv6 312894      0t0  TCP [200[redacted]:bc]:5228 (ESTABLISHED)
chrome    6331 bullseye   28u  IPv6 295599      0t0  TCP [200[redacted]01]:https (ESTABLISHED)
chrome    6331 bullseye   32u  IPv4 304846      0t0  TCP 192.[redacted]:https (ESTABLISHED)
chrome    6331 bullseye   34u  IPv6 312910      0t0  UDP [200[redacted]0.net:443
chrome    6331 bullseye   35u  IPv4 309428      0t0  TCP 192.[redacted]:https (ESTABLISHED)
chrome    6331 bullseye   37u  IPv4 311182      0t0  UDP 224.[redacted]
chrome    6331 bullseye   39u  IPv4 295595      0t0  TCP 192.[redacted]:https (ESTABLISHED)
chrome    6331 bullseye   42u  IPv4 299066      0t0  TCP 192.[redacted]
chrome    6331 bullseye   47u  IPv4 299883      0t0  TCP 192.[redacted]:https (ESTABLISHED)
chrome    6331 bullseye   49u  IPv4 303156      0t0  TCP 192.[redacted]https (ESTABLISHED)
chrome    6331 bullseye   51u  IPv4 308278      0t0  TCP 192.[redacted]et:https (ESTABLISHED)
chrome    6331 bullseye   73u  IPv4 224458      0t0  TCP 192.[redacted]
chrome    6331 bullseye   82u  IPv4 249264      0t0  TCP 192.[redacted]https (ESTABLISHED)
chrome    6331 bullseye  122u  IPv6 250356      0t0  TCP [200[redacted]-0000-0000-0000-11a6.deploy.st
atic.ak[redacted].com:https (ESTABLISHED)
┌─[bullseye@parrot]─[~]
└──╼ $
```

strace

strace is a powerful command-line tool for debugging and troubleshooting. It captures and records all system calls made by a process and the signals received by the process.

If **strace** is not pre-installed on your Linux system, run the appropriate command below for your distribution, to install it.

Debian/Ubuntu

Code

```
sudo apt install strace
```

RHEL/CentOS

Code

```
yum install strace
```

Fedora 22+

Code

```
dnf install strace
```

Arch-based

Code

```
man strace
```

man strace Determine if your Linux computer or server has been hacked

Code

```
strace ls
```

Code

```
strace -d -p <PID Number>
```

strace ls

Using ps

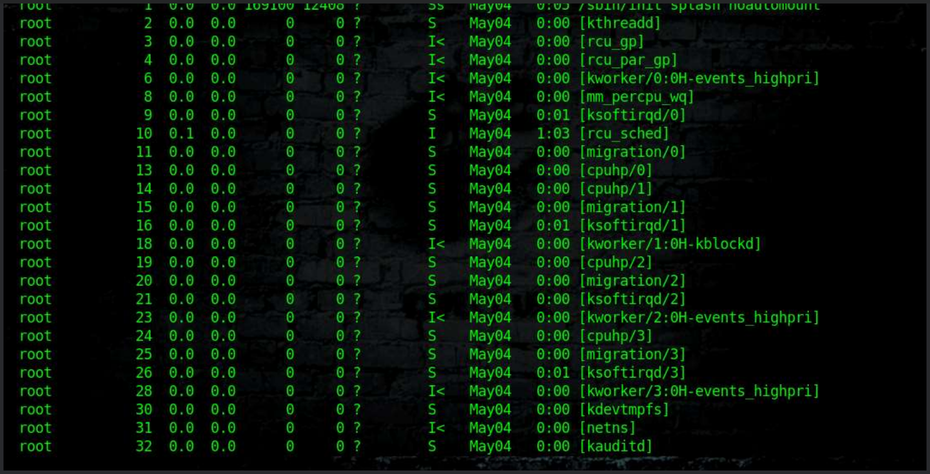
The **ps** (process status) command is one of the most frequently used commands in Linux. Usually it is used to get the more and detailed information about a specific process or all processes. For example it is used to know whether a particular process is running or not, who is running what process in system, which process is using higher memory or CPU, how long a process is running, etc.

Use the “man ps” for more info.

Code

```
ps aux
```

```
a = show processes for all users
u = display the process's user/owner
x = also show processes not attached to a terminal
```



```
root           1  0.0  0.0 169100 12408 ?        Ss   May04   0:05 /sbin/init splash noautomount
root           2  0.0  0.0      0     0 ?        S    May04   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   May04   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   May04   0:00 [rcu_par_gp]
root           6  0.0  0.0      0     0 ?        I<   May04   0:00 [kworker/0:0H-events_highpri]
root           8  0.0  0.0      0     0 ?        I<   May04   0:00 [mm_percpu_wq]
root           9  0.0  0.0      0     0 ?        S    May04   0:01 [ksoftirqd/0]
root          10  0.1  0.0      0     0 ?        I    May04   1:03 [rcu_sched]
root          11  0.0  0.0      0     0 ?        S    May04   0:00 [migration/0]
root          13  0.0  0.0      0     0 ?        S    May04   0:00 [cpuhp/0]
root          14  0.0  0.0      0     0 ?        S    May04   0:00 [cpuhp/1]
root          15  0.0  0.0      0     0 ?        S    May04   0:00 [migration/1]
root          16  0.0  0.0      0     0 ?        S    May04   0:01 [ksoftirqd/1]
root          18  0.0  0.0      0     0 ?        I<   May04   0:00 [kworker/1:0H-kblockd]
root          19  0.0  0.0      0     0 ?        S    May04   0:00 [cpuhp/2]
root          20  0.0  0.0      0     0 ?        S    May04   0:00 [migration/2]
root          21  0.0  0.0      0     0 ?        S    May04   0:00 [ksoftirqd/2]
root          23  0.0  0.0      0     0 ?        I<   May04   0:00 [kworker/2:0H-events_highpri]
root          24  0.0  0.0      0     0 ?        S    May04   0:00 [cpuhp/3]
root          25  0.0  0.0      0     0 ?        S    May04   0:00 [migration/3]
root          26  0.0  0.0      0     0 ?        S    May04   0:01 [ksoftirqd/3]
root          28  0.0  0.0      0     0 ?        I<   May04   0:00 [kworker/3:0H-events_highpri]
root          30  0.0  0.0      0     0 ?        S    May04   0:00 [kdevtmpfs]
root          31  0.0  0.0      0     0 ?        I<   May04   0:00 [netns]
root          32  0.0  0.0      0     0 ?        S    May04   0:00 [kauditd]
```

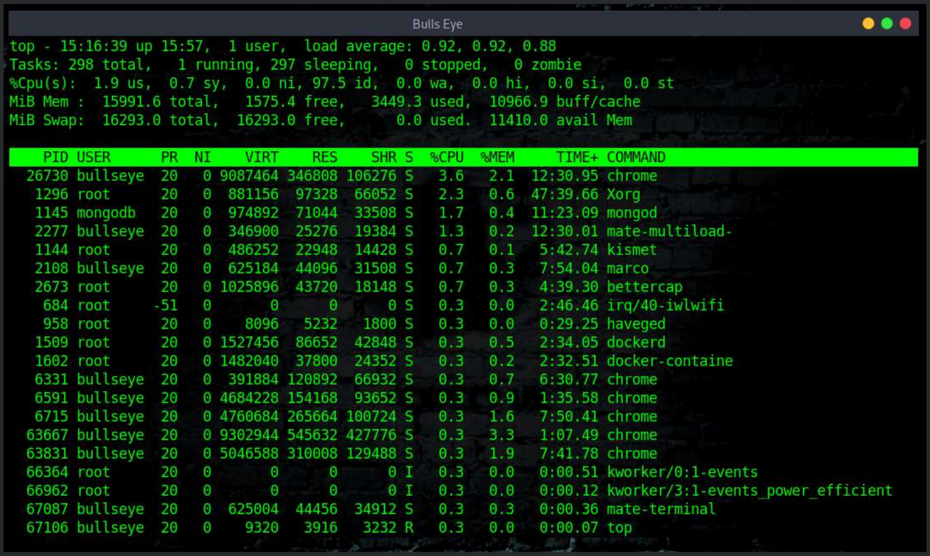
Check the running processes with TOP

The top command is a quick way to see what processes are consuming resources. **top** comes pre-installed on every Linux distribution. top it is interactive, and you can browse through the list of processes, kill a process, and so on. As you might have already guessed, you simply need to type this in to launch top.

You can use the arrow keys and Page Up/Down keys to browse through the list. If you want to quit, simply press "q".

Code

```
top
```

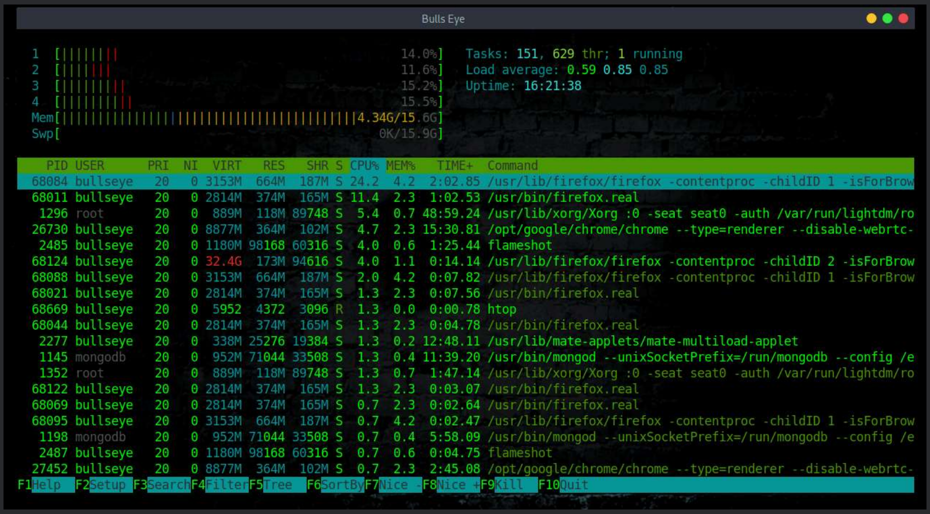


```
top - 15:16:39 up 15:57,  1 user,  load average: 0.92, 0.92, 0.88
Tasks: 298 total,   1 running, 297 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.9 us,  0.7 sy,  0.0 ni, 97.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15991.6 total,   1575.4 free,   3449.3 used,  10966.9 buff/cache
MiB Swap:  16293.0 total,  16293.0 free,      0.0 used.  11410.0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  26730 bullseye  20   0 9087464 346808 106276 S   3.6   2.1  12:30.95 chrome
   1296 root      20   0  881156  97328  66052 S   2.3   0.6  47:39.66 Xorg
   1145 mongodb   20   0  974892  71044  33508 S   1.7   0.4  11:23.09 mongod
   2277 bullseye  20   0  346900  25276  19384 S   1.3   0.2  12:30.01 mate-multiload-
   1144 root      20   0  486252  22948  14428 S   0.7   0.1   5:42.74 kismet
   2108 bullseye  20   0  625184  44096  31508 S   0.7   0.3   7:54.04 marco
   2673 root      20   0 1025896  43720  18148 S   0.7   0.3   4:39.30 bettercap
    684 root     -51   0       0      0      0 S   0.3   0.0   2:46.46 irq/40-iwlwifi
    958 root      20   0    8096   5232   1800 S   0.3   0.0   0:29.25 haveged
   1509 root      20   0 1527456  86652  42848 S   0.3   0.5   2:34.05 dockerd
   1602 root      20   0 1482040  37800  24352 S   0.3   0.2   2:32.51 docker-containe
   6331 bullseye  20   0  391884 120892  66932 S   0.3   0.7   6:30.77 chrome
   6591 bullseye  20   0 4684228 154168  93652 S   0.3   0.9   1:35.58 chrome
   6715 bullseye  20   0 4760684 265664 100724 S   0.3   1.6   7:50.41 chrome
  63667 bullseye  20   0 9302944 545632 427776 S   0.3   3.3   1:07.49 chrome
  63831 bullseye  20   0 5046588 310008 129488 S   0.3   1.9   7:41.78 chrome
  66364 root      20   0       0      0      0 I   0.3   0.0   0:00.51 kworker/0:1-events
  66962 root      20   0       0      0      0 I   0.3   0.0   0:00.12 kworker/3:1-events_power_efficient
  67087 bullseye  20   0  625004  44456  34912 S   0.3   0.3   0:00.36 mate-terminal
  67106 bullseye  20   0    9320   3916   3232 R   0.3   0.0   0:00.07 top
```

keyboard. When you first launch htop, you'll be greeted with a colorful interface showing a list of all processes running on the system. These are normally ordered by the amount of CPU usage, ordered from highest to lowest. It also shows the status of CPU usage, physical and swap memory.

Kill a Process Without Exiting From htop – Press F9 or k

To kill a process, Select the process that needs to be killed from the list, and press F9 or k, which will display the "Send signal" menu that lists all the available signals that you can send to the command.



Bulls Eye

```
  1  [|||||||||                                     14.0%]   Tasks: 151, 629 thr; 1 running
  2  [||||||||                                      11.6%]   Load average: 0.59 0.85 0.85
  3  [||||||||||                                    15.2%]   Uptime: 16:21:38
  4  [|||||||||||                                   15.5%]
  Mem[|||||||||||||||||||||||||||||||||||||||||4.34G/15.6G]
  Swp[                                          0K/15.9G]

    PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
  68084 bullseye   20   0 3153M  664M  187M S 24.2  4.2  2:02.85 /usr/lib/firefox/firefox -contentproc -childID 1 -isForBrow
  68011 bullseye   20   0 2814M  374M  165M S 11.4  2.3  1:02.53 /usr/bin/firefox.real
   1296 root       20   0  889M  118M 89748 S  5.4  0.7 48:59.24 /usr/lib/xorg/Xorg :0 -seat seat0 -auth /var/run/lightdm/ro
  26730 bullseye   20   0 8877M  364M  102M S  4.7  2.3 15:30.81 /opt/google/chrome/chrome --type=renderer --disable-webrtc-
   2485 bullseye   20   0 1180M 98168 60316 S  4.0  0.6  1:25.44 flameshot
  68124 bullseye   20   0 32.4G  173M 94616 S  4.0  1.1  0:14.14 /usr/lib/firefox/firefox -contentproc -childID 2 -isForBrow
  68088 bullseye   20   0 3153M  664M  187M S  2.0  4.2  0:07.82 /usr/lib/firefox/firefox -contentproc -childID 1 -isForBrow
  68021 bullseye   20   0 2814M  374M  165M S  1.3  2.3  0:07.56 /usr/bin/firefox.real
  68669 bullseye   20   0  5952  4372  3096 R  1.3  0.0  0:00.78 htop
  68044 bullseye   20   0 2814M  374M  165M S  1.3  2.3  0:04.78 /usr/bin/firefox.real
   2277 bullseye   20   0  338M 25276 19384 S  1.3  0.2 12:48.11 /usr/lib/mate-applets/mate-multiload-applet
   1145 mongodb    20   0  952M 71044 33508 S  1.3  0.4 11:39.20 /usr/bin/mongod --unixSocketPrefix=/run/mongodb --config /e
   1352 root       20   0  889M  118M 89748 S  1.3  0.7  1:47.14 /usr/lib/xorg/Xorg :0 -seat seat0 -auth /var/run/lightdm/ro
  68122 bullseye   20   0 2814M  374M  165M S  1.3  2.3  0:03.07 /usr/bin/firefox.real
  68069 bullseye   20   0 2814M  374M  165M S  0.7  2.3  0:02.64 /usr/bin/firefox.real
  68095 bullseye   20   0 3153M  664M  187M S  0.7  4.2  0:02.47 /usr/lib/firefox/firefox -contentproc -childID 1 -isForBrow
   1198 mongodb    20   0  952M 71044 33508 S  0.7  0.4  5:58.09 /usr/bin/mongod --unixSocketPrefix=/run/mongodb --config /e
   2487 bullseye   20   0 1180M 98168 60316 S  0.7  0.6  0:04.75 flameshot
  27452 bullseye   20   0 8877M  364M  102M S  0.7  2.3  2:45.08 /opt/google/chrome/chrome --type=renderer --disable-webrtc-
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Install NTop on Windows

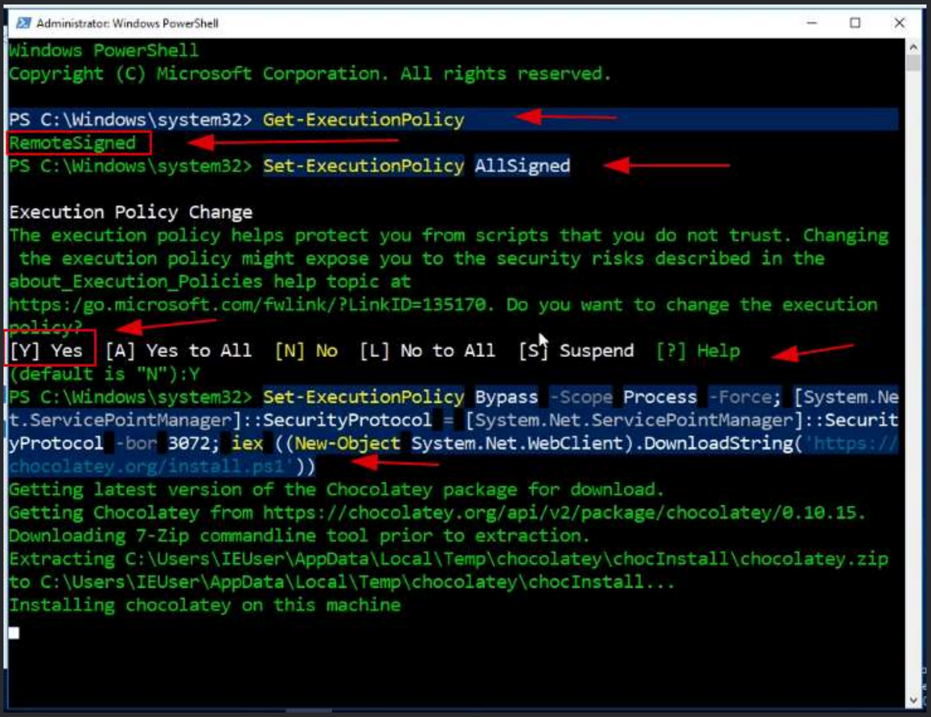
NTop is an Htop-like system-monitor with Vi-emulation for Windows.

Because using Task Manager is not cool enough 😜

First, you have to open in the search bar **PowerShell** and **run it as administrator**. Now we going to install chocolatey

Code

```
Get-ExecutionPolicy
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Get-ExecutionPolicy
RemoteSigned
PS C:\Windows\system32> Set-ExecutionPolicy AllSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing
 the execution policy might expose you to the security risks described in the
about_Execution_Policies help topic at
https:/go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution
policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help
(default is "N"):Y
PS C:\Windows\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Ne
t.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::Securit
yProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://
chocolatey.org/install.ps1'))
Getting latest version of the Chocolatey package for download.
Getting Chocolatey from https://chocolatey.org/api/v2/package/chocolatey/0.10.15.
Downloading 7-Zip commandline tool prior to extraction.
Extracting C:\Users\IEUser\AppData\Local\Temp\chocolatey\chocInstall\chocolatey.zip
to C:\Users\IEUser\AppData\Local\Temp\chocolatey\chocInstall...
Installing chocolatey on this machine
```

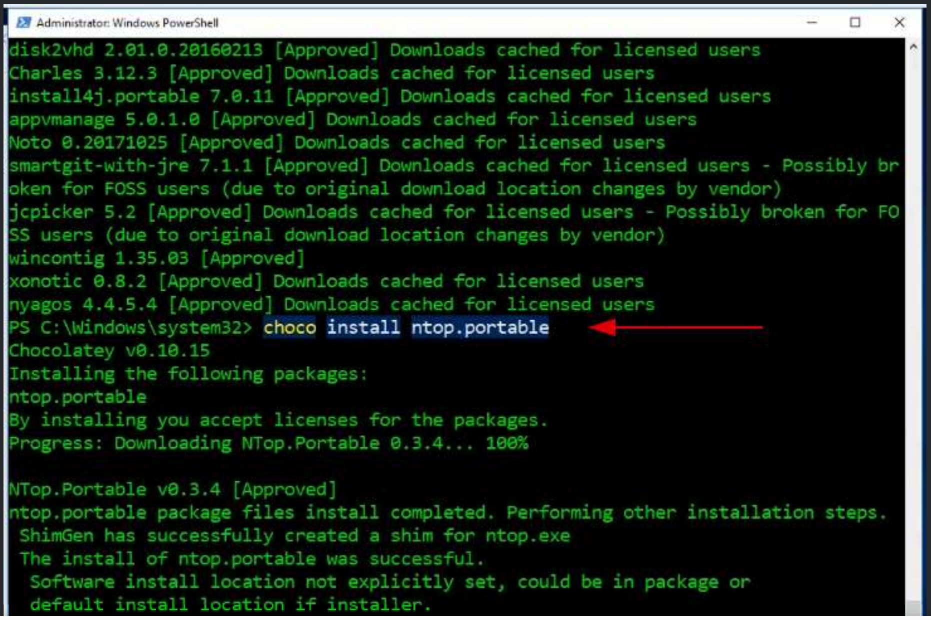
Code

```
choco -?
```

Now that it is installed, you can install NTop

Code

```
choco install ntop.portable
```



```
disk2vhd 2.01.0.20160213 [Approved] Downloads cached for licensed users
Charles 3.12.3 [Approved] Downloads cached for licensed users
install4j.portable 7.0.11 [Approved] Downloads cached for licensed users
appvmanage 5.0.1.0 [Approved] Downloads cached for licensed users
Noto 0.20171025 [Approved] Downloads cached for licensed users
smartgit-with-jre 7.1.1 [Approved] Downloads cached for licensed users - Possibly br
oken for FOSS users (due to original download location changes by vendor)
jcpicker 5.2 [Approved] Downloads cached for licensed users - Possibly broken for FO
SS users (due to original download location changes by vendor)
winconfig 1.35.03 [Approved]
xonotic 0.8.2 [Approved] Downloads cached for licensed users
nyagos 4.4.5.4 [Approved] Downloads cached for licensed users
PS C:\Windows\system32> choco install ntop.portable
Chocolatey v0.10.15
Installing the following packages:
ntop.portable
By installing you accept licenses for the packages.
Progress: Downloading NTop.Portable 0.3.4... 100%

NTop.Portable v0.3.4 [Approved]
ntop.portable package files install completed. Performing other installation steps.
 ShimGen has successfully created a shim for ntop.exe
 The install of ntop.portable was successful.
  Software install location not explicitly set, could be in package or
  default install location if installer.
```

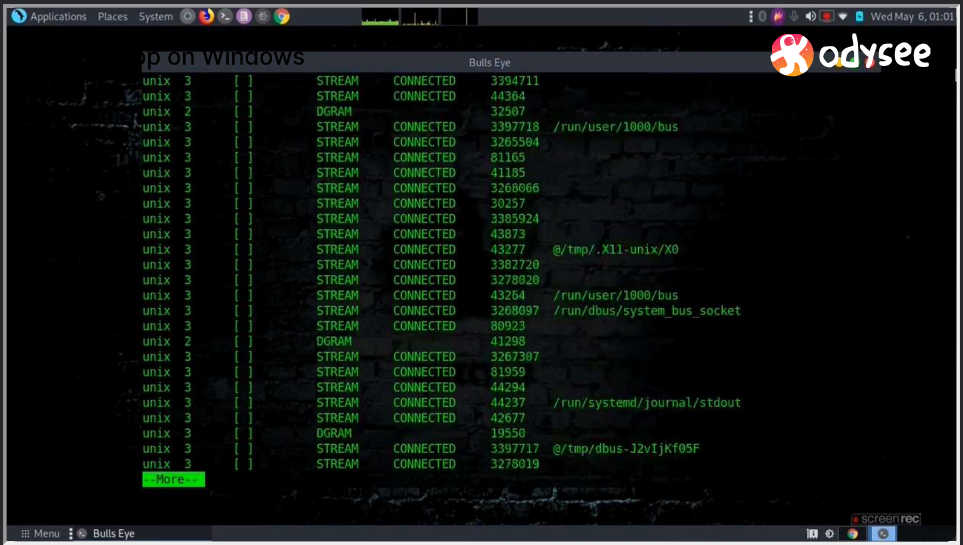
```
ntop
```

Using NTop for Windows

https://hackingpassion.com/wp-content/uploads/2020/05/

Video NTop on windows

In this video, I show you how to install NTop on Windows. The last part of the video shows how to use KILL “**k**” and shows you **nstat | more**.



```
unix  3      [ ]         STREAM     CONNECTED     3394711
unix  3      [ ]         STREAM     CONNECTED     44364
unix  2      [ ]         DGRAM                    32507
unix  3      [ ]         STREAM     CONNECTED     3397718  /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     3265504
unix  3      [ ]         STREAM     CONNECTED     81165
unix  3      [ ]         STREAM     CONNECTED     41185
unix  3      [ ]         STREAM     CONNECTED     3268066
unix  3      [ ]         STREAM     CONNECTED     30257
unix  3      [ ]         STREAM     CONNECTED     3385924
unix  3      [ ]         STREAM     CONNECTED     43873
unix  3      [ ]         STREAM     CONNECTED     43277    @/tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     3382720
unix  3      [ ]         STREAM     CONNECTED     3278020
unix  3      [ ]         STREAM     CONNECTED     43264    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     3268097  /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     80923
unix  2      [ ]         DGRAM                    41298
unix  3      [ ]         STREAM     CONNECTED     3267307
unix  3      [ ]         STREAM     CONNECTED     81959
unix  3      [ ]         STREAM     CONNECTED     44294
unix  3      [ ]         STREAM     CONNECTED     44237    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     42677
unix  3      [ ]         DGRAM                    19550
unix  3      [ ]         STREAM     CONNECTED     3397717  @/tmp/dbus-J2vIjKf05F
unix  3      [ ]         STREAM     CONNECTED     3278019
--More--
```

Become a member on **Odysee.com**
Earning on Odysee for watching videos ♥
Here an invitation link, so that we both benefit.
In this way, you also support my work.

https://odysee.com/$/invite/@hackingpassion:9

Check SSH attempt connections

Check the SSH logs to understand if somebody is trying to get access to the server, or computer.
You can check the access log to the server (SSH) in this way.

This command will show you the log from the last 300 lines of all the attempts

Tip: If you need to read backward the log you need to increase the number of lines to 1000 or more, depending on the server use because of this logfile store all access to the server (FTP, SSH, Webmin, and other...)

If you are using a Debian distribution based

Code

```
tail -n 300 /var/log/auth.log
```

Code

```
tail -n 300 /var/log/auth.log | grep sshd
```

If you are using a Centos/RedHat distribution based

Code

```
tail -n 300 /var/log/secure
```

Code

```
tail -n 300 /var/log/secure | grep ‘sshd’
```

You can use the top command to see what happens on your own PC. The numbers are adjustable.

Check SSH attempt connections

Open ports

Which ports do you have open? you can see this very well with nmap. A simple nmap scan will do for an initial overview.

Code

The proc file system is a pseudo-file system which provides an interface to kernel data structures. It is commonly mounted at /proc. Read the man page for more info. man proc

Code

```
ls /proc/*/exe -la
```

or:

Code

```
sudo ls /proc/*/exe -la
```

ls /proc/*/exe -la

Common attack points

These are all the common unsecured places where the hacker intrudes into your Linux machine

Code

```
ls /tmp -la
```

Code

```
ls /dev/shm -la
```

Code

```
ls /var/tmp -la
```

Common attack points

Crontab scheduled jobs

Another way is to check the **cronjobs**. Maybe a malicious script or application could be seen here.

“The **crontab** is a list of commands that runs on a regular schedule. Crontab stands for “**cron table** ” because it uses the job scheduler cron to execute tasks.”

less /etc/crontab Determine if your Linux computer or server has been hacked

How to view /etc/crontab on Linux

Code

```
cd /etc/
```

Code

```
ls -l
```

Determine if your Linux computer or server has been hacked

View Software Specific Cronjobs

Code

```
cd /etc/cron.d/
```

Code

```
ls -l
```

Code

```
cat filename
```

cd /etc/cron.d/ls -l

Listing users cron jobs when using systemd timers

Systemd comes with cron system called systemd.timer. This is another option that you can use on systemd based Linux distro. Use the systemctl command as follows to list cron jobs in Linux

Code

```
systemctl list-timers
```

Listing users cron jobs when using systemd timers

Conclusion