

How to Stop and Disable Unwanted Services from Linux System

www.tecmint.com/remove-unwanted-services-from-linux/

We build a server according to our plan and requirements, but what are the intended functions while building a server to make it function quickly and efficiently. We all know that while installing a Linux OS, some unwanted Packages and Application gets installed automatically without the knowledge of a User.

When building a server we need to ask ourselves what we actually need from the box. Do I need a **Web Server** or a **FTP Server**, a **NFS Server** or a **DNS Server**, a **Database Server** or something else.

Here in this article, we will be discussing some of these unwanted applications and services which you might not needed but they are installed by default during OS installation and unknowingly start eating your system resources.

Lets first know what kind of services are running on the system using the following commands.

```
[avishek@tecmint]# ps  
ax
```



Sample Output

```

PID TTY          STAT     TIME COMMAND
 2 ?           S         0:00 [kthreadd]
 3 ?           S         0:00 \_ [migration/0]
 4 ?           S         0:09 \_ [ksoftirqd/0]
 5 ?           S         0:00 \_ [migration/0]
 6 ?           S         0:24 \_ [watchdog/0]
 7 ?           S         2:20 \_ [events/0]
 8 ?           S         0:00 \_ [cgroup]
 9 ?           S         0:00 \_ [khelper]
10 ?           S         0:00 \_ [netns]
11 ?           S         0:00 \_ [async/mgr]
12 ?           S         0:00 \_ [pm]
13 ?           S         0:16 \_ [sync_supers]
14 ?           S         0:15 \_ [bdi-default]
15 ?           S         0:00 \_
[kintegrityd/0]
16 ?           S         0:49 \_ [kblockd/0]
17 ?           S         0:00 \_ [kacpid]
18 ?           S         0:00 \_
[kacpi_notify]
19 ?           S         0:00 \_
[kacpi_hotplug]
20 ?           S         0:00 \_ [ata_aux]
21 ?           S        58:46 \_ [ata_sff/0]
22 ?           S         0:00 \_
[ksuspend_usbd]
23 ?           S         0:00 \_ [khubd]
24 ?           S         0:00 \_ [kseriod]
.....

```

Now, let's have a quick look at the processes accepting connection (ports) using the [netstat command](#) as shown below.

```

[avishek@tecmint]# netstat -
lp

```

Sample Output

```

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 *:31138                 *:*
LISTEN    1485/rpc.statd
tcp        0      0 *:mysql                 *:*
LISTEN    1882/mysqld
tcp        0      0 *:sunrpc                 *:*
LISTEN    1276/rpcbind
tcp        0      0 *:ndmp                   *:*
LISTEN    2375/perl
tcp        0      0 *:webcache               *:*
LISTEN    2312/monitorix-http
tcp        0      0 *:ftp                     *:*
LISTEN    2174/vsftpd
tcp        0      0 *:ssh                     *:*
LISTEN    1623/sshd
tcp        0      0 localhost:ipp            *:*
LISTEN    1511/cupsd
tcp        0      0 localhost:smtp           *:*
LISTEN    2189/sendmail
tcp        0      0 *:cbt                     *:*
LISTEN    2243/java
tcp        0      0 *:websm                  *:*
LISTEN    2243/java
tcp        0      0 *:nrpe                   *:*
LISTEN    1631/xinetd
tcp        0      0 *:xmltec-xmlmail        *:*
LISTEN    2243/java
tcp        0      0 *:xmpp-client            *:*
LISTEN    2243/java
tcp        0      0 *:hpvirtgrp              *:*
LISTEN    2243/java
tcp        0      0 *:5229                   *:*
LISTEN    2243/java
tcp        0      0 *:sunrpc                 *:*
LISTEN    1276/rpcbind
tcp        0      0 *:http                   *:*
LISTEN    6439/httpd
tcp        0      0 *:oracleas-https        *:*
LISTEN    2243/java
....

```

In the above output, you notice that some of the applications you may not needed on your server but they are still running as follows:

1. smbd and nmbd

smbd and nmbd are daemon of Samba Process. Do you really need to export smb share on windows or other machine. If not! why are these processes running? You can safely kill these processes and disable them from starting automatically when machine boots the next time.

2. Telnet

Do You need bidirectional interactive text-oriented communication over internet or local area network? If not! kill this process and turn-off it from starting at booting.

3. rlogin

Do you need to log in to another host over network. If not! Kill this process and disable it from starting automatically at boot.

4. rexec

The Remote Process Execution aka rexec lets you execute shell commands on a remote computer. If you don't require to execute shell command on a remote machine, simply kill the process.

5. FTP

Do you need to transfer files from one host to another host over Internet? If not you can safely stop the service.

6. automount

Do you need to mount different file systems automatically to bring up network file system? If not! Why is this process running? Why are you letting this application to use you resource? Kill the process and disable it from starting automatically.

7. named

Do you need to run NameServer (DNS)? If not what on earth is compelling you to run this process and allow eating up your resources. Kill the running process first and then turn-off it from running it at boot.

8. lpd

lpd is the printer daemon which makes it possible to print to that server. If you don't need to print from the server chances are your system resources are being eaten up.

9. inetd

Are you running any inetd services? If you are running stand alone application like ssh which uses other stand alone application like Mysql, Apache, etc. then you don't need inetd. better kill the process and disable it starting next time automatically.

10. portmap

Portmap which is an Open Network Computing Remote Procedure Call (ONC RPC) and usages daemon rpc.portmap and rpcbind. If these Processes are running, means you are running NFS server. If NFS server is running unnoticed means your system resources are being used up un-necessarily.

How to Kill a Process in Linux

In order to kill a running process in Linux, use the '**Kill PID**' command. But, before running Kill command, we must

know the **PID** of the process. For example, here I want to find a PID of '**cupsd**' process.

```
[avishek@tecmint]# ps ax | grep cupsd
1511 ?          Ss      0:00 cupsd -C
/etc/cups/cupsd.conf
```

So, the PID of '**cupsd**' process is '**1511**'. To kill that PID, run the following command.

```
[avishek@tecmint]# kill -9
1511
```

To find out more about kill command with their examples, read the article [A Guide to Kill Command to Terminate a Process in Linux](#)

How to Disable a Services in Linux

In **Red Hat** based distributions such as **Fedora** and **CentOS**, make use of a script called '**chkconfig**' to enable and disable the running services in Linux.

For example, lets disable the Apache web server at the system startup.

```
[avishek@tecmint]# chkconfig httpd off
[avishek@tecmint]# chkconfig httpd --del
```

In **Debian** based distributions such as **Ubuntu**, **Linux Mint** and other Debian based distributions use a script called **update-rc.d**.

For example, to disable the Apache service at the system startup execute the following command. Here '**-f**' option stands for force is mandatory.

```
[avishek@tecmint]# update-rc.d -f apache2
remove
```

After making these changes, The system next time will boot without these UN-necessary process which in-fact will be saving our system resource and the server would be more practical, fast, safe and secure.

That's all for now. I'll here we again with another interesting article. Till then stay tuned and connected to Tecmint. Don't forget to provide us with your valuable feedback in comment Section.