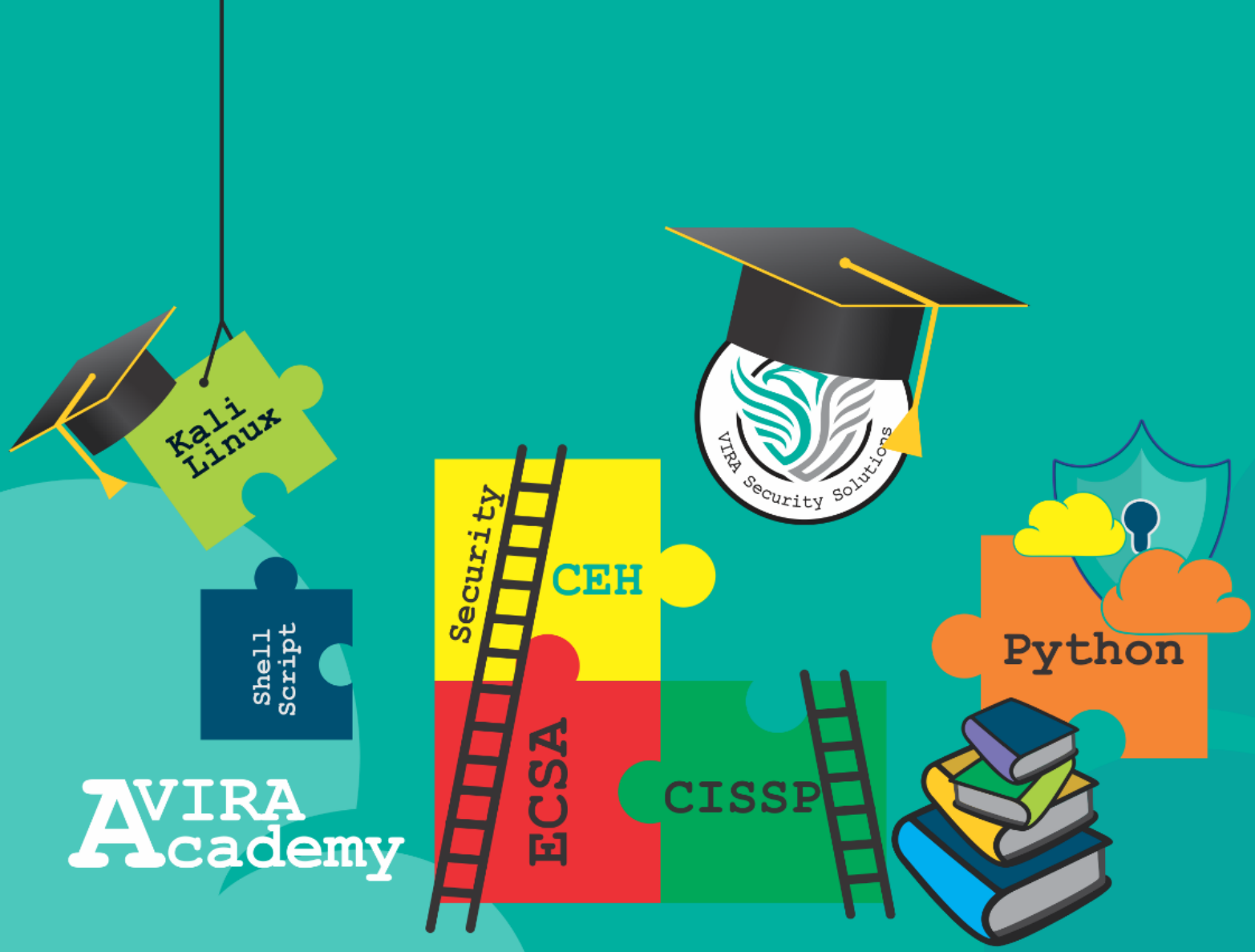

VIRA
Security
Solutions


راهکارهای امن ویرا
شماره ثبت: ۵۴۰۱۳۹
VIRA Security Solutions


Kali Linux
Shell Script
VIRA Security Solutions
Security
CEH
ECSA
CISSP
Python
AVIRA Academy

Linux

Part7

Mohammad Reza Gerami
Mrgerami@aut.ac.ir
gerami@virasec.ir
April 21 2020

Linux Tips

Tips

How to Disable Shutdown and Reboot Commands in Linux

The shutdown command schedules a time for a Linux system to be powered down, it may as well be used to halt, power-off or reboot the machine when invoked with particular options and reboot instructs the system to restart.

Certain Linux distros such as Ubuntu, Linux Mint, Mandriva just to mention but a few, make it possible to reboot/halt/shutdown the system as a normal user, by default. This is not ideal setting especially on servers, it must be something to worry about especially for a system administrator.

In this article, we will show how to disable shutdown and reboot commands for normal users in Linux.

Tips

Disable Shutdown and Reboot Commands in Linux

The easiest way to disable shutdown and reboot commands using the /etc/sudoers file, here you can specify a user (vira) or group (developers) which are not allowed to execute these commands.

Vi /etc/sudoers

```
Cmnd_Alias     SHUTDOWN =
/sbin/shutdown,/sbin/reboot,/sbin/halt,/sbin/poweroff

# User privilege specification
vira   ALL=(ALL:ALL) ALL, !SHUTDOWN

# Allow members of group sudo to execute any command
%developers  ALL=(ALL:ALL) ALL,  !SHUTDOWN
```

Tips

Disable Shutdown and Reboot Commands in Linux

Another way is to remove execution permissions on shutdown and reboot commands for all users except root.

```
# chmod o-x /sbin/shutdown
# chmod o-x /sbin/reboot
```

Note: Under systemd, these file(/sbin/shutdown, /sbin/reboot, /sbin/halt, /sbin/poweroff) are only symbolic links to /bin/systemctl:

```
# ls -l /sbin/shutdown
# ls -l /sbin/reboot
# ls -l /sbin/halt
# ls -l /sbin/poweroff
```

Tips

Disable Shutdown and Reboot Commands in Linux

To prevent other users from running these commands, you would simply remove execution permissions as explained above, but this is not effective under systemd. You can remove execution permissions on /bin/systemctl meaning all other users except root will only run systemctl.

```
# chmod  o-x /bin/systemctl
```

Tips

How to Stop and Disable Unwanted Services from Linux System

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.

Tips

How to Stop and Disable Unwanted Services from Linux System

ps ax

```
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]
```

Tips How to Stop and Disable Unwanted Services from Linux System

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

netstat -lp

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

Tips How to Stop and Disable Unwanted Services from Linux System

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.

```
ps ax | grep cupsd

1741 ?        Ss     0:00 cupsd -C /etc/cups/cupsd.conf
```

So, the PID of ‘cupsd‘ process is ‘1741‘. To kill that PID, run the following command.

```
kill -9 1741
```

Tips How to Stop and Disable Unwanted Services from Linux System

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.

```
chkconfig httpd off
chkconfig httpd --del
```

Tips How to Stop and Disable Unwanted Services from Linux System

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.

```
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.

Tips

Sudoers

user_list host_list= effective_user_list tag_list command_list

user_list – list of users or a user alias that has already been set.
host_list – list of hosts or a host alias on which users can run sudo.
effective_user_list – list of users they must be running as or a run as alias.
tag_list – list of tags such as NOPASSWD.
command_list – list of commands or a command alias to be run by user(s) using sudo.

To allow a user (vira in the example below) to run all commands using sudo without a password.

Tips

Sudoers

user_list host_list= effective_user_list tag_list command_list

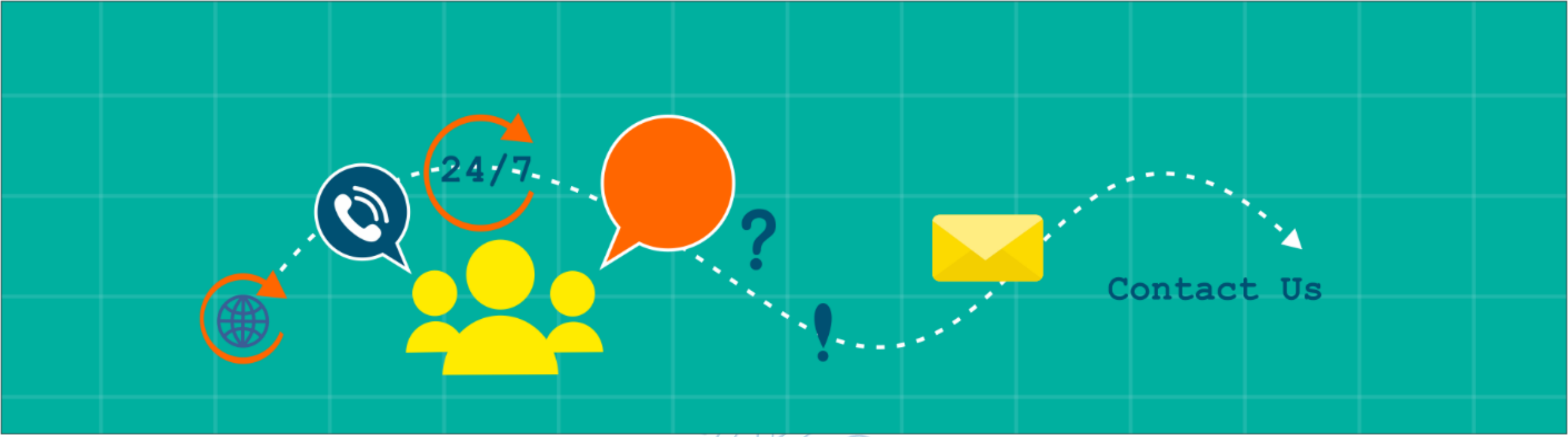
visudo
And add the following line:

vira ALL=(ALL) NOPASSWD: ALL
For the case of a group, use the % character before the group name as follows; this means that all member of the sys group will run all commands using sudo without a password.

%sys ALL=(ALL) NOPASSWD: ALL
To permit a user to run a given command (/bin/kill) using sudo without a password, add the following line:

vira ALL=(ALL) NOPASSWD: /bin/kill
The line below will enable member of the sys group to run the commands: /bin/kill, /bin/rm using sudo without a password:

%sys ALL=(ALL) NOPASSWD: /bin/kill, /bin/rm



Visiting Address: Unit 20, Floor 4, No 53 Vafa Manesh Ave

Heravi, Pasdaran Ave, TEHRAN-IRAN

Post Code:1668838803

Tel No: 0098 21 2298 1027-09125792641

Email: info@ virasecsolutions.com

Website: www.virasecsolutions.com

آدرس: تهران، پاسداران، هروی، خیابان وفامنش، پلاک ۵۳

طبقه چهارم، واحد ۲۰

کد پستی: ۱۶۶۸۸۳۸۸۰۳

شماره تماس: ۰۹۱۲۵۷۹۲۶۴۱-۰۲۱۲۲۹۸۱۰۲۷