* **Linux troubleshooting** :-

1. **Grub file Password Reset**
2. **Root Password Reset**
3. **No Root Login problem**
4. **Recover deleted inittab file**
5. **Recover missing initrd file**

* **Networking :-**

1. Network Configuration & Troubleshooting
2. Network(NIC)/Ethernet/ Bonding / Teaming
3. Firewall/IPTABLES

**=====================================================================================**

* 1) **Grub file Password Reset** :-

GRUB is a multiboot boot loader for linux system. User can select OS to run from menu interface displayed when a system boots up.

Here we will cover kernel panic error and grub password

* **kernel panic error** : while booting if getting below error, it means it is boot loader related problem.

Kernel Panic - not syncing: Attempted to kill init!

We can pass the kernel parameter from the boot loader as well as we can correct the kernel parameter passing from boot loader from GRUB screen at boot time.

Steps :-

* Reboot system and press space bar on boot menu and select kernel line
* Now press e for edit and try to see wrong entry of kernel line in file and correct it.

Note :- This will correct this error temporary so after booting system correct the kernel parameter

* **Grub password :**

Two types of password can be set on grub.conf one to edit the parameter in grub.conf during boot process and another to boot operating system.

Set password for editing just below the hiddenmenu and for booting the OS just below the title

**Hiddenmenu title CentOS**

**Password 123 Password 222**

Then restart the system. It will ask password.

* Now we lost all 3 passwords.
* To enter into **Rescue mode :**

Boot system from Linux CD and give linux rescue command on boot screen

In Rescue mode system will be mounted on /mnt/sysimage folder and then we can make any changes required.

To enter into root env have to run #chroot /mnt/sysimage

Finally, open Grub file and remove both entry

* 2) **Root Password Reset :-**

Protecting password can enhance the security of system.

To change password even if single user mode is password protected.

* At the time of booting press any key to edit the grub menu.
* Select the OS and **press e**
* Select the kernel line and **press e again**
* Change kernel parameters so write **init=/bin/bash** OR [space]**init 1** and press enter to come out of this.
* Now press b to boote.

This is to tell kernel that instead of starting normal inittab file just give a simple bash shell.

Note :- Need to mount / partition in RW mode **#mount –o remount,rw /**

* **Restrict access to single user mode :**

Make entry “**su:S:wait:/sbin/sulogin”** in the **/etc/inittab** file just above the **initdefault** line

* 3) **No Root Login problem** :-
* 4) **Recover deleted inittab file** :-

Init is the first process running under linux and this file is used to start init process. This file also start runlevel and if deleted we won’t be able to boot. We can’t even use single user mode.

Steps-

* Will have to enter into **rescue mode** and enter into root env.
* Mount CD or iso file using **#mount /dev/sda /mnt**

Note :- To know system path to CD or iso use **df –h**

* Install **initscripts rpm** file stored under server folder of the CD or iso.

**# rpm –Uvh –force /mnt/server/init---**

* **5) Recover Missing initrd image file :-**

This is the file needed during booting to initialize the kernel. Even booting into single user mode doesn’ t work.

Steps :-

* Boot into Rescue mode and try to recover
* Open **/boot/grub/grub.conf** file and make the changes
* List /boot contents and file name gets alter then **rename it** to original name using **mv** command**.**

OR

We can recreate this image by using

**# mkinitrd -o /boot/initrd.$(uname -r).img $(uname -r).**

Also, need to modify **grub.conf** file to point correct ramdisk image

The line should be present **initrd /boot/initrd.img-2.6.x.x.img**

* **Networking :-**

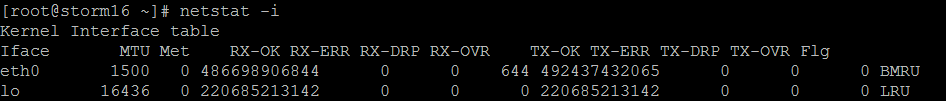
1. **Network Configuration & Troubleshooting :-**

The basic requirements for n/w are

1. **NIC (Network interface card)/Network adapter, LAN adapter**

Is a computer hardware component that connects a comp. to a comp. network. Each NIC have a unique MAC(media access control) address to avoid conflicts between same NIC adapters and represented by **“eth”**.

* To show network interface packet transactions including transferring and receiving with MTU size **#netstst –i**



* **ethtool eth0/lo** - Display or change ethernet card settings
* **To change IP address :**
* #**setup** or #**system-config-network**
* Restart the network service

If changes not reflected, then restart **#service NetworkManager restart**

1. **Media**

It is a cable via two different computer’s NIC card will be connected.

1. **Topology**

Is the scheme or design in which computer in the network will be connected.

Ex: Bus, Ring, Star, Mesh, Tree

1. **Protocol**

Defines rules for communication between network devices. Protocol use packet switching techniques to send and receive messages in the form of packets.

|  |  |
| --- | --- |
| **TCP** | **UDP** |
| Transmission Control Protocol | User Datagram Protocol |
| It is connection Oriented | Connectionless |
| Reliable | Non-Reliable |
| **TCP Acknowledgement will be sent/received** | **No Acknowledgement for UDP** |
| Slow Communication | Faster Communication |
| Protocol Number for TCP is **6** | Protocol Number for UDP is **17** |
| **HTTP, FTP, SMTP** uses TCP | **DNS, DHCP** uses UDP |

1. **Subnet Mask/netmask :-**

It allows us to identify which part of IP address is reserved for the **network** and which part is available for **host** use. Also, we can calculate whether IP address are on the same subnet or not.

Commanly used netmask is a 24-bit.

1. **Gateway :-**

It is a network point that **provides entrance to another network** of different networks**.** The computers that control traffic within network or at local internet service provider(ISP) are gateway nodes.

* **To get info. of default gateway** :

**# netstat –r OR route –n** will display routing table

# **ip route | grep default**

* Add / setup a new route

**Syntax : # route add default gw {IP-ADDRESS} {INTERFACE-NAME}**

**#route add default gw 192.168.1.254 eth0**

* **Networking troubleshooting :-**

N/W conf. files determine whether networking is started during boot process or not.

- > In trouble of network we should first check

1) **# /etc/init.d/network status** OR **# service network status**

It list configured and active devices.

Note:- If we made any change in IP conf. then restart n/w interface

1. Next is to get into the conf. files

**# cat /etc/sysconfig/network**

This file contains global configuration settings also, specifies **routing** and **host** info. for all n/w

If directive **NETWORKING** is set to **no** then **/etc/init.d/network** script doesn’t activate any network devices. It must be **set to yes** in order to start networking.

If networking still not start check the status of networking service. Then check status of network service, it should be active in appropriate runlevels.

**#** **chkconfig --list network**

network 0:off 1:off **2:on 3:on 4:on 5:on** 6:off

Note:- If service network restart command return error for eth0 then try to reactivate particular network device with **ifdown** or **ifup** commands.

* **What is port** ?
* A port is piece of software, where remote application can communicate. We can see all port information in **/etc/services** file.
* Port numbers can vary from **0 to 65535**, this is because limitation in TCP/IP where port number is **16 bit size**. So we can get only 2^16 ports available.

In that assigned(default,system) ports are from **0 to 1023(2^10)** which is designated port for particular server**.**

**1024-49151** are registered or user ports

& other is dynamic or private ports

* We can change default port for a service. In apache & DNS we can change this using

Listen configuration entry in httpd.conf file.

* **What is socket ?**

Combination of software **port+IP address and protocol(TCP/UDP)** and every service must have a unique socket

Form of interprocess communication, but socket can pass data and info. between two process that are not running on the same machine

Ex – web browser use socket to make connection to the web server

* **To list open/listening ports** or all network connections :

for finding a connection to and from the host.

We need to know which all open ports, to check if particular ports is listening on server or not, particular service is communication on configured port or not

#**netstat –natup**l

Options :-

- **l :** listening sockets/connections

**-n :** show port number

**-a** : shows all ports(tcp/udp)

**-p** : show **Process ID** and **process name** of application which is using that port

**-t/-u** : show TCP/UDP connection



#**lsof –i**

Using this we can trace current open ports



We can see which command run by which user using which socket and state of port

* To check if the port is associated with known services **#/etc/services | grep portnumber**
* Most reliable way is **nmap –sT –o localhost**
* **To check port status on Remote host :-**

**#nmap host/IP** finding **open ports, services, and MAC address** on remote server

**#nmap -A IP** To detect OS info. Running on remote host

**#nmap –sP IP.\*** To check which hosts are live and up in network

**\* # nmap –p 80 IP/Host** To scan for specific port on remote server

* **#ss –lntu**

**> #last** command shows how many users are logging with their terminal, Remote IP and time.

Proxy/squid server :- To provide direct web access(DNS not required) and can handle internet connection between a browser and internet sites directly

1. **Network(NIC)/Ethernet/ Bonding / Teaming :-**

Bonding enables 2 or more NIC card to single virtual NIC card. If one physical NIC is down, it will automatically move resources to other NIC card.

It is used where we cannot afford to lose connectivity due to Ethernet failover and also have advantages like distribute bandwidth, provides redundancy, load balancing, fault tolerance etc.

There are 6 types of bonding.

0:**Load balancing**(Round-Robin) : Traffic is transmitted in round-robin fashion from both NIC. Provides load balancing.

1:**Active-Backup** : Only one slave NIC card is active at any point of time. Other interface will be active only when active NIC fails

======== Create Bonding ==========

1. Configure **eth0** and **eth1** :

#MASTER=bond0

#SLAVE=yes

1. Create **ifcfg-bond0** file in the /etc/sys./net-scripts

**DEVICE-bond0**

IPADDR=

Network/Gateway=

NETMASK=

**BONDING\_OPTS=”mode=0 miimon=100”**

Note:- miimon=100 means polling intervals

1. Make sure bonding module is loaded into kernel /**etc/modprobe.con**

**Alias bond0 bonding**

**Options bond0 mode=balance-alb miimon=100**

And execute module **#modprobe bonding**

1. Restart network service

1. **Firewall/IPTABLES :-**

Firewall is simply a host whose main purpose is to **protect network**. It restricts certain types of network traffic from the internet.

Need of Firewall-

* Increase network security
* Network access control
* Examine all Inbound/Outbound network traffic

About iptables -

Iptables is a command-line firewall utility that uses to allow or block network traffic.

**Types of chains** :-

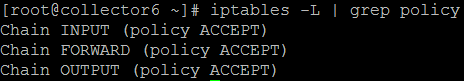
**Input**- This chain is used to control the behavior for **incoming** connections. For incoming **SSH** connections, iptables will check its input chain to match IP address and port to a rule.

**Forward-**

**Output-** This chain is used to control the behavior for **outgoing** connections. To **Ping**, iptables will check its output chain before making decision to allow or deny the connection attempt.

* **Policy chain Default Behavior**

To decide/check default behavior of 3 chains. What iptables want to do if the connection doesn’t match any existing rules.



Our chains are currently figured to accept traffic

Command to accept/deny all connections by default

**#iptables --policy INPUT/OUTPUT/FORWARD ACCEPT/DROP**

* **Connection-specific Responses**

**Accept** – Allow the connection

**Drop**- Drop the connection. When we don’t want the source to realize our system exists



**Reject-** Don’t allows the connection, but sends back an error. When we don’t want the source to connect to our system, but want them to know that firewall blocked them.



With default policy chain configured, we can configure iptables to allow or block specific address or ports

- > To block all connection from IP 10.10.10.10. or we can use netmask or std slash notation

**# iptables -A INPUT -s 10.10.10.10/255.255.255.0 -j DROP**

-> To block SSH connection from IP

**# iptables -A INPUT -s IP -p tcp --dport ssh/22 -j DROP**

**Note :-** If we want to block SSH from any IP then don’t specify **–s(Source)** option

To allow multiple ports

**#iptables –A INPUT –p tcp –m multiport --dports 22,80,442 -j ACCEPT**

- > Allow N/W range on port 22

**# iptables -A OUTPUT -p tcp -d IP --dport 22**

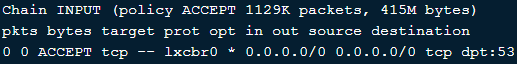
- > To unblock use **# iptables –D INPUT**

* **Connection States**

**Saving changes**

**/sbin/service iptables save OR /etc/init.d/iptables save**

* To clear all the currently configured rules use **# iptables –F**
* To check existing rules **# iptables -L -n -v**



# **iptables -A INPUT -s 0/0 -i eth0 -d 192.168.1.1 -p TCP –j ACCEPT**

iptables is being configured to allow the firewall to accept TCP packets coming in on interface eth0 from any IP address destined for the firewall's IP address of 192.168.1.1. The 0/0 representation of an IP address means any.

Where, A – Append rules to the existing chain

s - source IP address

i - “Input” interface on which the packet enters

d – Destination IP Address

p – protocol(icmp, TCP, UDP) tells what kind of connection

-p tcp/udp --sport/--dport <port> Source/Destination port

j - ACCEPT/DROP

- > To forward one service traffic to another port.

# **iptables –t nat -A PREROUTING -i eth0 -dport 25 -p TCP –j REDIRECT –-to-port 2525**

-> To add ftp service to the firewall in RHEL7.

**#firewall -cmd --permanent --add-service=ftp**

**firewall-cmd --permanent --zone=public --add-port=80/tcp**

On RHEL7 :

**#firewall-cmd --add-permanent –add-rich-rule=’rule family=“ipv4” source address=“172.168.x.x” --add-service=“ssh” –accept/reject’**