4/1/2020

Unit - 1

→ Advantages of Linux OS :

→ Multi processor	- flexible	GNU GPL
- Multi platform	- Stable	
Interoperable	- efficient	
scalable	- free	General public
Portable	- Multiuser	license

1) Significants of GNU, GPL, Beneolandal dictator live CD distribution, Core distribution, Specialized dist'.

* Distribution : Packages
  1) Core eg redhat, fedora, debian
  2) live CD : eg : nopix, pc linux os, puppy, slax
  3) Specialised distribution. eg : Zendros, ubuntu, puppy, linux

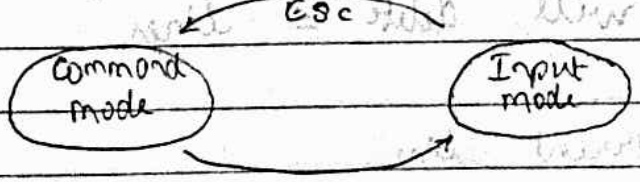
8/1/2020

Unit 2 VI - Editor

Visual editor
Vim Improved

- Vi editor : Visual editor - current version vim
- hard to learn - Case sensitive - powerful
- vi <filename> - 2 modes :- 1) Command mode
- Originally written by : Bill Joy
- press "Esc" : opens command mode.

2) Input mode :



Insert (i, I)
append (a, A)
open (o, O)
replace (r, R)

→ moving cursor position:

h - for left
l - for right
j - for down
k - for up
w - 1 word forward
b - 1 word backward
<enter> - next line.
H - current screen Home position
L - moves lower last line
M - moves middle line on the current screen
0 - moves cursor to 1st character in the current line.
control - d - scrolls down
control - u - scrolls up
control - f - ——"—— forward
control - b - ——"—— backward
x - deletes current character
dw - deletes end of word.
dd - deletes current line
d$ - deletes to beginning of line

The vi editor commands can be used followed number such as

n < command key (s) >

eg 5dd : will delete 5 lines

yy - copy current line
nyy - n no. of lines copy

p - paste lines below cursor
P - ——— // ——— above ——— //
u - undo changes
. - repeat last edit command.
ZZ - save & exit , :w - save
:q — <enter> is to exit, search - /keyword

10/1/2020

touch - change time stamp / create files
tail - to display last line // tail = 5 ②
where is - find location & display // whereis tail
df - print files with memory
uname - will display all data // uname - a
shutdown / -r reebod
who am i - gives current user login info.
ping - to check connection.
rpm - packet install, update. is done.
mount - to find mounting point
chkconfig - list of process // /grep - give sorted
ntconfig -

touch [opt] [filename]
-a → access time is changed
-m → modified time is changed.
-am → access time & modified time is changed
-r → reference file

uname -a all info
uname -p process info

uname -s Kernel info
uname -k version
uname -m machine.

13/01/2020

→ Linux Booting :
Ⓐ / Power On
1) startup H/W /init/ — BIOS / (CPU → BIOS) →
POST → 1st boot/ device

→ MAN → manual → gives total command.

2) tail
tail filename
→ last 10 lines will be printed

tail 5 filename
→ last 5 lines will be printed

to find location of file

~ where is filename
tail [top] devak
↓
will print directory

~ df → will print all files with disk file memory

~ df -h
↳ will print files storage in mb.

uname → kernal info is printed

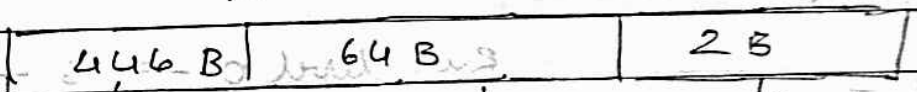
V.Imp

→ Booting

Power ON :

1) Startup / H/W initialization — BIOS (CPU → BIOS)
→ POST → 1st Boot device

2) BL → Boot loader stage1 — local MBR
512 B
master boot record



contains executable code & error msg.

446 B: primary boot loader code
locator
↓
store location of actual boot loader code

64 B: partition Info
size file system, mount, point, etc.
4 partition mask

2 B: Magic bit - To retrive file if corrupted
↳ for validating MBR

Boot loader stage 2

3) GRUB → Grand unified boot loader

- stored in 30 kb after MBR
- load boot menu
- load default kernel (if we don't select the kernel)

4) Kernel
→ start init process
→ reside until shutdown in RAM

rc ⓓ → demons file

5) Init process

→ root process — fork() is called

→ etc/rc.d & etc/init.d

1st file → etc/rc.d/rc.sysinit

/etc/rc.d/rc.sysinit → load Kernel modules

1st process executed by Init → file system mount

/etc/inittab = defines starting of sys program.

2nd file — etc init tab → system process table

Run level 0 → 6 → total 7

/etc/rc.local — last command in initialization stage

etc/rc0.d → etc/rc6.d

last → etc/rc.local

Control → kernel

shutdown

halt file → in etc/init.d

* Diff between LILO & GRUB

c) user prompt

getty process — getty

→ gives login prompt

MBR - MASTER BOOT RECORDER

→ LILO — linux loader [Sept 2015]
discontinued

→ 16 boot img
→ not specific file system
→ inside MBR

lilo.conf file →
→ global option → boot location
→ Boot action option
→ image

L → No lilo
LI → bad disk
LIL → media failure to load map file
LIL ? → 2nd stage addr incorrect
LIL description table corrupted
LILO → loaded successfully

*	LILO	GRUB
1)	16 images at a time	Unlimited no. of boot entries
2)	must be written again every time you change the conf file	~~LILO~~ Windows.
3)	Cannot boot from nw	can boot from network
4)	Don't have interactive command interface	has

* GRUB working : BIOS → "Real boot" → active partition → boot

replace MBR with self-code

stage 1 → point to 2nd.

stage 1.5 → continue → usr into

stage 1.5 → if small 20 - 30 kb

sys V init process

— sequential

— newer : upstat & systemd

— AT&T

etc/inittab file run by sbin/init process

sysin

0 - halt, 1 - single, 6 - Reboot

sbin/initx *

* For red hat linux :

2 - not defined

3 - multi user without graphics (textmode)

4 - user defined

5 - multi user with gui

* File system :

Used in linux.

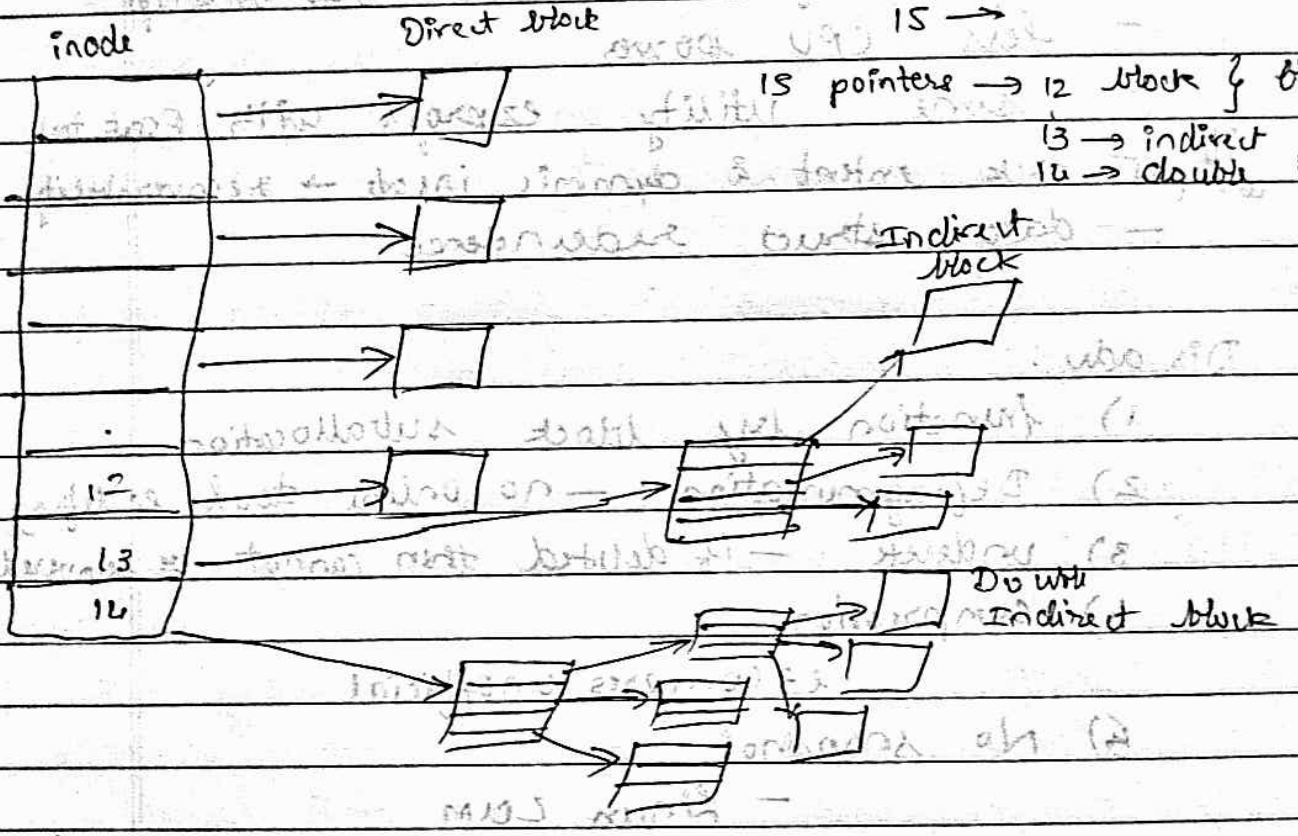
ext - extended file system.

ext2, ext3, ext4 (now).

* Superblock, block groups, descriptor table, block bitmap

OS Booting

inode bitmap inode table → actual blocks



18/01/2020

* Inodes :

- contains size, permission ownership
- 12 for direct blocks

Around 30,000 subdirectories

Directory :

- dirl entries → associated filename with inode num
- search inefficient
- ext3 HTree
- root dir at inode 2

ext3

ext3 - performance less than JFS VFS
HTree hai
- upgrade from ext2 with abt backup
Journaling hai
- less CPU power
- same utility e2progs with fsck tool
- lock extent & dynamic inode → recoverability
if corrupted
- data struct redundence

Dis adv:
1) function by block suballocation
2) Defragmentation — no online tool e2defrag
3) undelete — if deleted then cannot be recovered
4) compression
e3 compres unofficial
5) No snapshot
- Linux LVM

20/01/2020

* VFS : Unix file system
- Beakly Fast FS/FFS
- Begining block : boot block
- Super block : magic no.
- Cylindrical block contains Backup of SB
- inodes
- data block

- cylinder group header

- Indodes Sequence - 0 - unallocated dir
  1 - bad block file
  2 - root dir
  3 - lost & found dir

RFS :- Reiser FS ; 1st journaling file system
- Nemesys teaten by Hans Reiser
- 2.4.1 linux : Introduced ; 1st to be introduced in kernel
- 1st Journaling
- default on Elive xandros ~~dist~~ distribution

* Features :
  metadata, Online Resizing , Tail Packing

→ JFS 64 bit by IBM (Journaling FS)
- versions workstation AIX, OS/2 & linux distribution

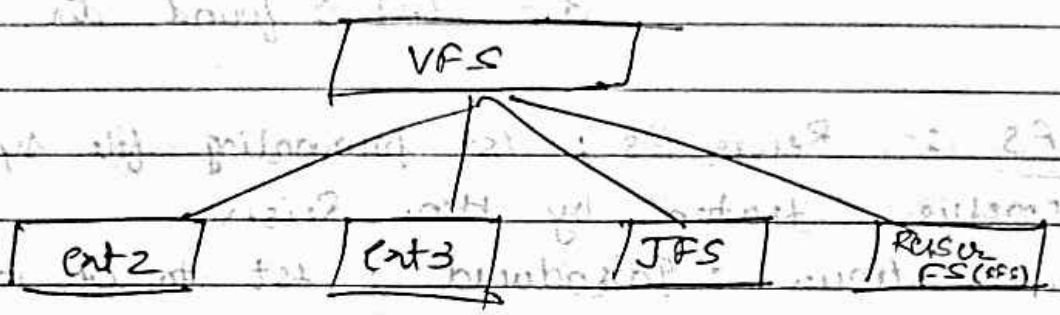
* features:
  1) Journal : max 128 mb
  2) B+ tree
  3) Dynamic inode allocation : 512 bytes
  4) Extents
  5) Compression : only in JFS1 & AIX
     LZ algorithm (single workstation)

→ Tail packing : reduces internal fragmentation (feature of RFS) :

6) CIO : ~~read~~ read shared / write exclusive location

7) allocation group
8) JFS superblock :- size, data block, flags.

* VFS : Virtual File System



→ Similar to registry in windows

23/01/2020 Unit 4

* Meta devices
- group of physical slices (logical device) appears as single LD to system
- creating methods :-
1) concat 2) striping 3) mirroring
4) RAID5 5) VFS : logging
(main block syst)

Types : 1) concatination 2) stripes 3) Concatination & stripes
4) mirrors 5) Raids metadevices 6) Transmission metadevices

- Disk suit (uses) → metadevices disk drives - Coordinate I/o
- enable meta → physical
- disksuits GUI, CMD Line Interface

Types:

1) Simple - Single building block
   - no redundancy

2) Mirrors : replicates multiples copies
   - submirrors.

3) Raid 5 :- Raplicates using parity information

4) Trans : Used to log as UFS FS
   - master & logging devices
   - Master → UFS FS

→ partition are called slices

1) concatination 2) stripes 3) Concatination stripes are basic building blocks

In Raid 5 whole data is not replicated

/dev/md/dsk/d0 → d0
/dev/md/dsk/d7

0 - 127 max 1024

Kernel/dev/md.cnf file

RAID - Redundant Array Independent Disk

8/2/2020

RAID :

- appear as on device
- / boot on raid1 array
- excellent performance

Raid0 : Striping { (segment to accomodate file on multiple drives)

- no redundancy
- faster access
- treat all hard drive as single RAID device
- use all storage. 80 + 80 = 160 GB (Not a replica)

Raid 1 : Mirroring

- inefficient
- 80+80 ⟶ 80 GB (Not a replica)

Raid 4 : Parity

- parity information on separate drive
- not supported in large distribution

Raid 5, 6 : distributed parity

- parity information equivalent to one drive
- min 3 drives
- Stripping + parity in Raid 5
- dual set of parity in Raid 6
  - fast access + recovery capability

LVM : logical volume manager for linux kernel

- Single volume of multiple drive.
- Similar to Raid 0
- allow resizing

- take snapshot of logical volum.
- use extents

Striping : the files are partioned & store in diff/ multiple harddisk.

* Xserver client – X windows system
- xserver program runs on local m
- handles graphic card i/o device, display
- X windo system → X
- X - cross platform, free, client server system for managing GUI on single or n/w computer
- Typical client server scenario
  local * local / remote n/c

- X → client – server invoked appl
- local – X server → access to client `local/ remote
  input/output
- X server – manages i/p display
- socket communication

Adv :
- No need to know low level n/w details to programmers
- simplify & facilitates on multiple computers
- uses Xprotocol for communication between X client & X server

Message format : 1) Request 2) Reply 3) Events
4) Error packet

Five basic services
1) I/p handling 2) Window service
3) Graphics 4) Text / Font
5) Resource Management - Database

Disadv: does not provide management for active window at any given time

3/02/2020

* Window Manager :-
- Coordinate all win on screen
- how to share and receive i/p
- part of X Windows Sys API provide size, pos etc.
- Actually win. man maintain real size position. wind overlay etc. & also i/p receiving win

- Hooks for man win - hierarchy
  - sub wind - buttons - events
  allow prog to manage size
- It add title bar to main window
- determines visibility in win overlay
- 2 responsibility - ① screen's colormap
  ② focus management - access to mouse & keyboard

* Runlevel in linux:
- designates different conf & allows access to different sets of services

- 0 : halt 1 : single user - rarely used
2 : multiuser - No network
3 : multiuser - command line
4 : user def
5 : multi user : GUI → desktop
6 : Reboot

(2, 3, 4, 5 → default; user can modify)

- for altering runlevel → prog is init & telinit command

/sbin/runlevel → runlevel/commands path

id:5:initdefault: → etc/inittab file.

query of runlevel are to troubleshoot any problem user as you forgot the user username etc

* Red Hat Packet Manager :- (RPM)
Open source under GPL
- info of installed package under /var/lib/rpm database
- deals with .rpm files
Commands : install, remove, upgrade, verify, query.

- package managment utility for Red Hat Dev. system (fedora centos etc)

It is a package magn^n tool/utility.

apt - for installing in ubuntu:

old versions of OS: [apt-get:
[apt-cash:

18/02/2020

→ Root account :-

- Root user, superuser, root account
- By default has access to all fields and commands other meanings.
- Root directory — `/`

→ Root directory :- Root user's home dir
- save users fields & conf. file
- open when logged in

/home dir :- home directory for other users.
- sub dir of /root

- Root Privilages - modify system in any way to grant or revoke permission.
- Rootkit → hacking tool to get access of root acct.
Root prog also has root permission → hackers exploit error's progs.

- Root only has write permission
- Mainframe - double terminal - admin & every simultaneous user

- User account UID → root account UID → ZERO

echo $UID

→ root can be dangerous to be used all the time so.

su, sudo, kdesu → provide root privileges without logging as root.

→ no virtually safety net for root user in case of careless error.

* sudo → provide ability run commands as root privilage
  - prepend sudo before commands
  - password prompt → cache for 5 min
  - command are logged
  - controlled access to user for staff they need without getting root password.

* User prompt groups:
  - created when new user is added
  - set default permission for newly created files called umask & configured in the file /etc/bashrc file.

default umask — 022

* Shadow Password :-
Traditionally password stored at /etc/passwd
— word - readable form
- stored as encrypted
— for authentication comparision between user password
encrypted by using key (salt) [which is 1st 2 char of user passwd]

- dictionary attack vulnerability

- shadow password :-
encrypted password → at /etc/passwd.
actual passw → along with expiration_date
kept in file read by root
- Root level permission to access that file
- increase security

* File System Quotas-
- limited amount of disk space / number of files user may allocate per system basis.
- prevent from consuming all space.
- control size → limit amount of size & no. of inodes
- 1 inode per files

20/02/2020

* Job scheduler :

Crom demon -

Crontab -e : to open

or sudo crontab -e

or su -c crontab -e

1) /bin/ed
2) /bin/nano ⟶ GNU Nano
3) /usr/bin/vim.tiny export editor=Nano

min (0-59) hour (0-23) day (1-31) month (1-12) weekday (0-6) . command location

29 0 * * * /usr/bin/ example

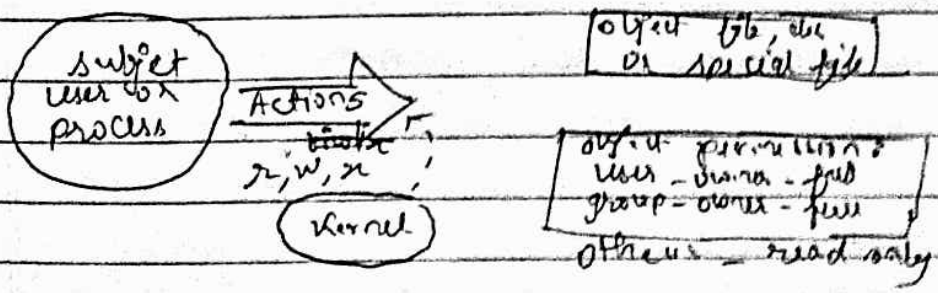
for every 15 min in an hour. 0,14,29,49 * * * * /usr/bin/example2

Crontab : installing new crontab.

24/02/2020 Linux system security (Unit 5)

- game of "root takes all"
- add on tools - sudo & tripwire

Discretionary Access Control (DAC)



* File system security

- use everything as file
- files, dir, devices, memory etc

user & group → main group
→ other group

group membership → /etc/password
→ addition group → /etc/group

1st field → user name
2nd → x - password
3rd → uid numeric
4th → gid numeric

- to modify file → useradd, user modify & user del

* File permission processing:

mautro
-rw- rw- r-- user 3544
own group other Mar 25 01:36 baton.txt

chmod → to change permission.

* Directory permission :

Read → list

write → add/delete file

Execute → we any thing within/ change to working dir by using cd command

Numeric file permission :

0700 →

file → 4 → read 2 → write 1 → execute

$2^1 - 1$

$2^1 - 2$

$2^2 = 4$

special permission 4 → set uid

2 → set gid

1 → sticky bit

0644.

25/02/2020

* PAM : Playable Authentication Module

→ Common frame work for Authentication & security

— Centralized authentication mechanism

— playable modular architecture

useful to ~~two of for~~ developer

Advantages :-

— common authn scheme used in vasity of appli

— flexibility to developer & system admin own authn.

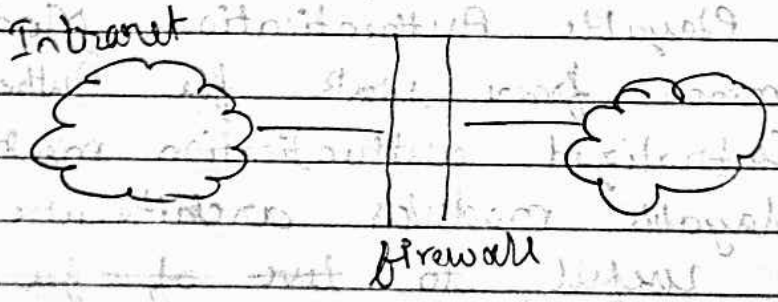
— provide fully documented library

— no need to create own authn scheme for developer

(Network address translation)

PAM conf files :-
in etc/pam.d dir
etc/pam.d/login - login
—"— /gdm - graphical login
—"— / samba - samba server

* Firewalls :-
- provide security to local n/w system from o/s based security through out allowing access to outside world.
- outer security wall / checkpoint where security and audit is taken place.
— monitoring security related event locations
= NAT & Internate usage audits and logs.
— VPN (platform)



dis Adv :

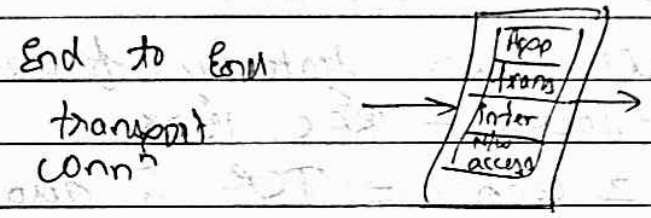
- not for bypassing system like dial up
- Internal threat
- not for wireless comm
- interested devices connected internally

26/02/2020

* Firewalls :-

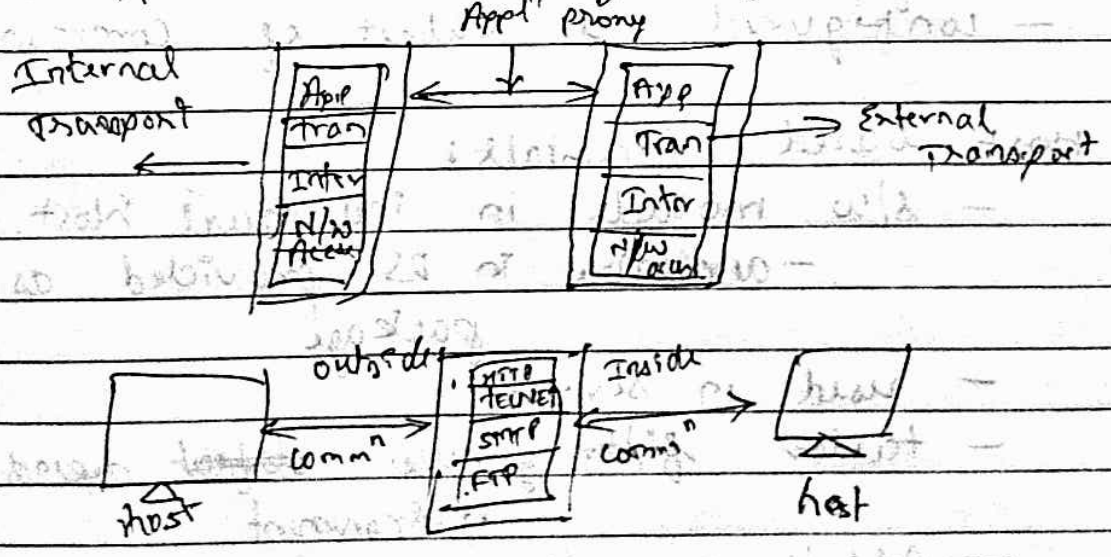
1) Packet filtering :

— Rules for incoming - outgoing packet
no match either → discard or allow
— rules based on info" in packet
- source & destination IP, port, transport protocol etc
- simplicity, flexibility, transparency speed.



* Stateful Inspection Packet Filter Firewall
— general firewall → no content of hyper layer
— TCP conn" above 1024 to 65335 allowed
— stateful → make dictionary of outbound
TCP conn"
— Keep TCP sequence no. → prevent session hijack

2) Application Level Gateway :-



- relay application level traffic
- contact remote host and relay all data between two endpoints like authentication, service
- proxy code for specific application
- more secure than packet filter
-

3) Circuit level gateway firewall
   - socks - RFC 1928
   - 2 conn^n - TCP & also UDP - 11 TCP and UDP ports.

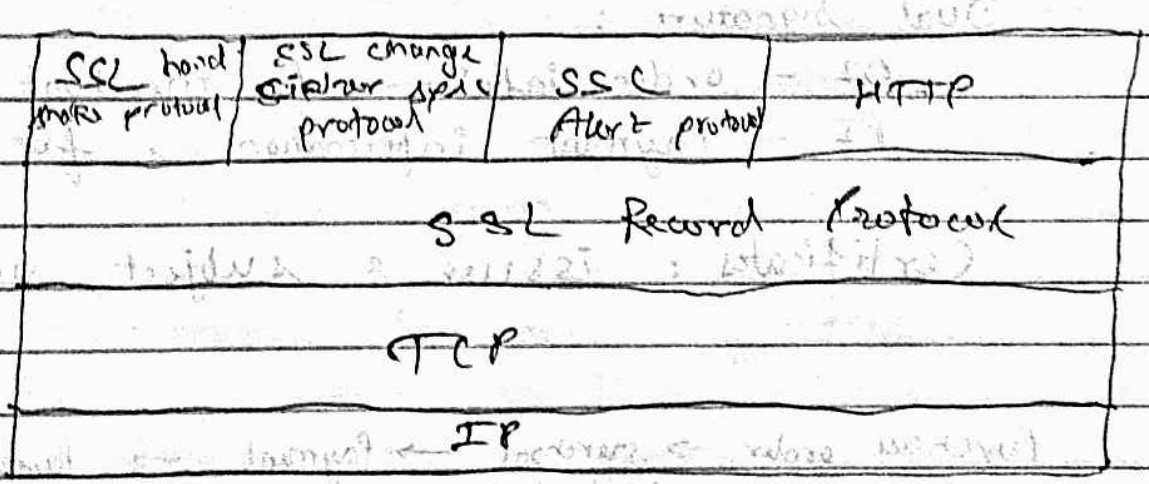
4/03/2020

* Bastion Host firewall - stand alone
  - runs as circuit level gateway
  - implimented as s/w ~~switch~~ module in router/switch
  - execute secure versions of OS.
  - only essential services installed
  - configured to subset of commands

* Host based Firewall:
  - s/w module in individual Host.
  - available in OS provided as a add-on package
  - used in server
  - tailor filtering rules ~~default~~ depend upon environment
  - Additional layer of security restricts flow of packets

* Personal firewall:
- Traffic between workstation or PC → enterprise or internet.
- S/w module on PC
- Home, office or DSL/cable.
- deny remote access to PC
- monitor addjoining activity → logging
- much less complex than server firewall.

* SSL - Secure Socket Layer
- Works in transport layer
- developed by netscape
- subsequently becomes standard as TLS.
- 2 layer of protocol.



Architecture → session & connection created by hand shake

SSL Record protocol:
Confidentiality & message integrity
Handshake encryption MAC with secret key
IDEA, DES 40 bit.

Change cipher spec :
- updating cipher suit

SSL Alert protocol :
Severity : warning or fatal

Specific alert : decomp failure, bad cert, record
expired no cert, unexpected msg

Handshake protocol - allow authentication

5/03/2020

* Electronic transaction in SSL

Dual Signature :
OI - order info → for merchant
PI - Payment information → for Bank

Certificates : issuer & subject, subject public key, time

Purchase order → merchant → Payment → Payment request by merchant
verifies merchant certificate
verify client certificate & check integrity
gateway Authorization
Transfer to merchant account
confirm order & dispatch goods

5/03/2020

Unit 6

* Samba Server :
  - for sharing and access between linux and window system & vice versa
  - access filesystem & printers
  - manage methods of authentication.
  - accounts with samba username & password.
  - ~~tbd~~ tdb → trivial database to store username & password.
  - to edit ~~pbd~~ tdb edit command
  - /etc/samba/ samba.conf file
  - GNOME & KDE to browse on Linux
  - my network places to browse on win → linux samba server
  - smb : URL

11/03/2020

* DNS
  - host name, domain name, subdomain
  - ~~Protocol~~ Initially local file
  - Name server software :
    → Berkeley Internet Domain (BIND) server
    → queried by resolvers

* MTA
  mail transfer Agents : SMTP - 25

mail access Agents → POP3 - 110
IMAP

Mail transfer agent:
→ Send mail: —
→ postfix — fast, easy to configure
→ gmail — flexible, fast & secure
→ Exim - MTA based on email
→ Courier

12/03/2020

* Apache Web Server:
— developed by apache s/w foundation (decentralized developers
— free HTTP web server s/w
— Based on NCSA server by National center for Supercomputing Appln.
— Originally for linux-unix → now crossplatform
— Manual is provided along with s/w

Features:
1) loadable Dynamic Modules
2) Hanling of static files index files etc
3) .ht access per-directory file configuration → dir in html sites
4) support https 2
5) gzip for compression & decompression
6) support ftp connect
7) used by many companies like TSM, FB etc
8) Support scripts file php

9) open source & customized do need by coding

Cons:
- Ability to modify opens threat & bugs
- performance issues on heavy traffic
  → tomcat apache.

* NFS:

NFS Daemon:
- operates over TCP/IP network.
- configuration file /etc/exports

→ NFS server access & deny controlled by etc/hosts.allow & etc/hosts.deny

daemons:

rpc.nfsd — receives request from NFS system & translate it to local system.

rpc.mounted — perform requested mount & unmount open.

rpc.portmapper — maps rimote request to appropriate daemon.

rpc.quotad — user disk quota managment.

rpc.statd: provide lock when remote host reboot.

rpc.lockd — handle lock recovery.

NFS configuration:

mount point & list of host access file system export options seprated by comma.

* → for any host

syn: dir pathname host - dsignation (options)

options:

secure — default

insecure -

ro — default

rw — read write

sync — write when requested

asyc - write all when rows ready.