

Unix Command Line Basics

Linux System Administration
Fall 2023

bash command line

- Default shell on modern Linux
 - Editing, history, autocompletion
- Built-in commands
 - if, for, while, case, select, alias, ...
- Parameter manipulation
- Shell variables
- Compound with | and ;

Bash startup Files

https://www.gnu.org/software/bash/manual/html_node/Bash-Startup-Files.html

- Interactive login
 - /etc/profile
 - First existing from
 - ~/.bash_profile
 - ~/.bash_login
 - ~/.profile
- Interactive non-login shell
 - ~/.bashrc
- Non-interactively
 - Defined by \$BASH_ENV
- Interactive login should explicitly source ~/.bashrc in profile
- Profile typically sets \$PATH and umask
- ~/.bashrc sets most things
 - prompt (\$PS1)
 - terminal (\$TERM)
 - aliases (~/.bash_aliases)
 - other variables as needed

ssh (secure shell)

- replaces telnet & rlogin/rsh
 - encrypted connection
 - basis for scp, rsync, etc
- Our go-to tool for access
 - Major goal of exams
 - ssh tunnel (-L and -R)
- Password-less login
 - `~/.ssh/authorized_keys`
- `ssh user@host`
- Server side sshd
- Default port 22
 - `p` for other port
- `ssh user@host 'command'`
- Forward X11 with `-X` or `-Y`
- Debug with `-vvvv`
- root logins may be disabled

Editing the command line

- Left and Right arrow, Home and End moves cursor
- Backspace and delete
- Tab completes file name
 - Can be context sensitive
- Ctrl-C aborts editing
- Up and down arrows scroll through the command history
- Ctrl-R searches command history for pattern
 - Ctrl-R searches for next
 - ESC to stop and edit
 - ENTER to stop and execute

Script arguments

- Positional parameters
 - `$0` name of script
 - `$n` parameter *n*
 - `$*` all parameters from 1
- Wildcard expansion is done by the shell
 - Script sees expanded
- Useful shell parameters
 - `$EUID` effective UID
 - 0 when run by sudo
 - `$PWD` current directory
 - `$HOSTNAME` host
 - `$LINES` window height
 - `$COLUMNS` window width

Compound commands

- Unix approach: commands should do one thing well.
Compound commands allow complex behavior
- Separate with ;

```
for file in *.txt; do mv $file $file.0; done
```
- Pipeline with |

```
awk -F: '{print $7}' /etc/passwd|sort|uniq -c
```
- Convert with xargs

```
grep -l -dskip foo *|xargs sed -i s/foo/bar/
```

One liners

- **Login shells in /etc/passwd**

```
awk -F: '{print $7}' /etc/passwd | sort | uniq -c
```

- **Do something on multiple hosts**

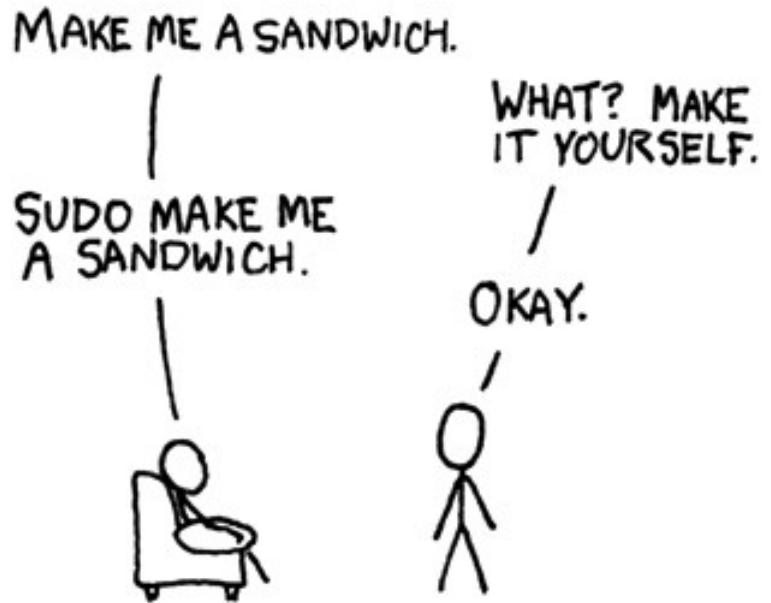
```
for host in foo bar; do ssh $host dnf install mlocate; done
```

- **Do something on specific files**

```
file * | grep CRLF | sed 's/: .*//' | xargs -l dos2unix
```

sudo

- Run command as root
 - Training wheels for root
 - Can limit commands
 - Provides audit trail
- /etc/sudoers
 - Group **sudo** or **wheel**
- ***USE IT!!!***
 - D.F.I.U.



<https://xkcd.com/149/>

Files and directories

- Everything is a file
- Unified directory tree
 - / is the directory separator
 - file systems mount as directories
- File names are arbitrary strings
 - White space in file names are a nightmare
 - Extensions are not definitive

Special file types

- directory - list of file names and inode pointers
 - hard link - directory entry to existing inode
- symbolic link - path to another file
- pseudo files
 - FIFO (named pipe)
 - socket
 - device file (block or character)

Where are stuff?

/boot – boot loader files

/etc – system configuration

/bin – essential binaries (`ls`, `cat`, `cp`, `mv`)

/sbin – essential system binaries (`mount`, `mkfs`, `fsck`, `modprobe`)

/usr – user utilities and files (`/usr/bin`, `/usr/share`, `/usr/lib`)

/var – variable files (`/var/log`, `/var/mail`, `/var/spool`)

/tmp – temporary files (cleaned out on boot)

/dev – device files

/proc – virtual system files (kernel access)

/home – user data

Manipulating the file system

- Directories
 - cd - change
 - ls - list
 - pwd - show current
 - mkdir - create new
 - rmdir - remove
- Files
 - cp - copy
 - mv - move
 - rm - remove
 - ln - link
 - cat - concatenate
 - scp - remote copy
 - rsync -remote sync

ls flags

- l long listing
- a also show dot files
- h show human format
- t sort by time
- S sort by size
- d show directory
- i show inode
- r reverse
- 1 list one file per line

- By default, files starting with . (period) are not shown
- List file by type
 - color color code
 - F append */=>@|
- Show files in human readable format ordered by time in reverse
 - ls -ltrh

Special directory names

- / root directory
- . current directory \$PWD
- .. parent directory
- ~ home directory \$HOME
- previous directory on cd \$OLDPWD

Moving files locally

- Copy file(s)

`cp from to`

- i prompt on overwrite
- a preserve attributes
- r recursive

Copy multiple files if
to is a directory

- Move file or directory

`mv from to`

- i prompt on overwrite

Preserves attributes by default

Move multiple files if
to is a directory

Copying files remotely

- `scp from to`
 - p preserve attributes
 - r recursive
 - P port
- `rsync from to`
 - a preserve attributes
 - r recursive
 - delete-excluded
- Layered on ssh
- Specify remote as
`user@host:dir`
- rsync copies only when changed files very efficiently

Locating files

- `find dirs`
 - Search directories
 - `name pattern`
 - `iname pattern`
 - Lots of options to find by type, time, permissions, etc.
- `which command`
 - Uses \$PATH to find executable for `command`
- `locate pattern`
 - Not installed by default
 - Database of files
 - `updatedb` scans filesystems

Finding out more about files

- `file`
 - Determines file properties by examining the contents of the file
 - Extensions are not definitive
- `stat`
 - Shows details of the file returned by `stat()` system call
 - Permissions
 - Disk usage
 - Times

Comparing files

- `diff file1 file2`
 - compare two files
- `diff -r dir1 dir2`
 - compare two trees
- `sdiff file1 file2`
 - show differences side-by-side
- `zdiff file1 file2.gz`
 - compare gzipped files

Useful diff flags

- q only report that files differ
- i ignore case
- b ignore changes in the amount of whitespace
- w ignore all white space
- Z ignore trailing whitespace
- B ignore blank lines

File permissions

- Owner:Group:Other
 - r - read (4)
 - w - write (2)
 - x - execute (1)
 - Directory x=list
- Example
 - `rwxr-xr--`
- `ls -l`
 - show permissions
- `chmod`
 - `chmod ug=rwx`
 - `chmod 775`
 - Values are octal
 - `chmod g+w`

Showing what is in a file

- cat – show all
- more - paginate
- head – first n lines
- tail – last n lines
 - f to follow changes
- grep pattern file(s)
 - lines with pattern
 - v to invert selection
- awk action file(s)
 - Show users in passwd
`awk -F: '{print $1}' /etc/passwd`

aliases

- Creates new or modifies commands
- Applies when it is the first word of a simple command
- Expanded by bash
- `~/.bash_aliases`
- `alias cp='cp -i'`
- `alias mv='mv -i'`
- `alias rm='rm -i'`
- `alias ll='ls -l'`
- `alias vi='vim'`
- `alias -p`
- `unalias ls`

Bash if and for

```
for x in *
do
  if [ -f "$x" ]
  then
    echo "$x is a file"
  else
    echo "$x is something else"
  fi
done
```

- Loop over all files
- If it is of type file
 - say it is a file
- else
 - say it is something else

if

Conditional action

```
if condition  
then  
    commands  
fi
```

Alternative actions

```
if condition  
then  
    commands  
else  
    commands  
fi
```

Nested decisions

```
if condition  
then  
    commands  
    if condition  
    then  
        commands  
    else  
        commands  
    fi  
else  
    commands  
fi
```

Cascading decisions

```
if condition  
then  
    commands  
elif condition  
then  
    commands  
else  
    commands  
fi
```

Conditional expressions

- `if [$x == 0]`
 - Is the string '0'
 - `if [$x -gt 0]`
 - Is \$x positive
 - `if ["$x" != abc]`
 - Is the string not abc
 - `if [-z "$x"]`
 - Is this a blank string
 - `if [-r foo.txt]`
 - Is foo.txt readable
- Hints
 - Beware string vs. numeric comparisons
 - Bash defaults to strings
 - Whitespace matters
 - Quote strings that may be blank
 - 0 means true
 - Lots more on man page
 - CONDITIONAL EXPRESSIONS

case

```
case $variable in
    pattern1)
        commands;;
    pattern2)
        commands;;
    patternN)
        commands;;
    *)
        commands;;
esac
```

- Hints
 - Comparison is based on string matching
 - Wildcards
 - * matches anything
 - Clauses are done in order
 - First match wins
 - No action if none match
 - Can be a convenient if

Iteration

- **for** var **in** list; **do** commands; **done**
 - Execute commands for values in list
- **while** cmd1; **do** cmd2; **done**
 - Do cmd2 while cmd1 returns true
- **until** cmd1; **do** cmd2; **done**
 - Do cmd2 until cmd1 returns true

Return values

- Generally 0 means everything is OK
 - This is just like most system function calls
- Return (exit) value of the last command is \$?
- 0 is true in bash
- Example: Can we ping 8.8.8.8?

```
ping -c 1 8.8.8.8
if [ $? ]; then
    echo "8.8.8.8 pings OK"
fi
```

Whitespace Matters

- Assignment

date=3 vs. date = 3

- Expressions

if [\$?];then vs. if [\$?];then

- Range

for x in {1..5} vs. for x in {1 .. 5}

Quotes matter

```
x=`date`  
for i in {1..3}  
do  
    echo "$X"  
    sleep 1  
done
```

```
X="date"  
for i in {1..3}  
do  
    echo `\$X`  
    sleep 1  
done
```

```
x=`date`  
for i in {1..3}  
do  
    echo '$X'  
    sleep 1  
done
```

```
X='date'  
for i in {1..3}  
do  
    echo `\$X`  
    sleep 1  
done
```

Bash arithmetic

```
x=5
```

```
y=6
```

```
z=$x+$y
```

```
echo $z
```

```
5+6
```

```
((z=$x+$y))
```

```
echo $z
```

```
11
```

- Hints:

- Bash treats variables as strings
- Double parentheses means it is an arithmetic expressions
 - ((z=\$x+\$y))
 - z=\$((\$x+\$y))
- Result is a string to bash

Scripts

- Comment start with #
- Initial line #! invokes an interpreter
 - #!/bin/bash - bash
 - #!/usr/bin/python3 - python 3
- File must be executable
- **You should get good at bash and another scripting language like Python or Perl**

bash arrays

- Initialize

```
days=( 'M' 'W' 'F' )
```

- Assign

```
days[2]='Saturday'
```

- Append

```
days+= 'Sunday'
```

- Access one element

```
echo ${days[1]}
```

- Access all elements

```
echo ${days[*]}
```

```
echo ${days[@]}
```

- Number of elements

```
echo ${#days[*]}
```

- Index of elements

```
echo ${!days[*]} }
```

- Declaration

```
declare -a days
```

- Associative arrays

```
declare -A name
```

Bash functions

- Definition

```
myfunc () { commands; }
```

or

```
function myfunc { commands; }
```

- Arguments are by position

- \$1 \$2 \$3

- Return value in \$?

- Use local to make variables local to function

Parameter expansion

- Isolate variable
 - `${variable}`
 - `dir/${file}a.txt`
- Substring
 - `${name:off:len}`
- Substitution
 - `FILE=foo.txt`
 - `${FILE/txt/pdf} = foo.pdf`
- Remove prefix
 - `${parameter#word}`
- Remove suffix
 - `${parameter%word}`
- Set default value
 - `${parameter:-value}`
- Parameter length
 - `${#parameter}`

Basic networking

- `ifconfig` configure interfaces
- `route` configure routing
- `ip` configure networking
- `ping` test connectivity
- `traceroute` test routing
- `dig` test DNS
- `nslookup` test DNS

Downloading files

- Get data from a web or ftp server
 - ftp supports wildcards
- wget
 - m is great for mirroring entire directory tree
- curl (cURL)
 - Supports many more protocols

Users and Groups

Linux System Administration
Fall 2023

What is a User Account?

- A user account is a set of credentials that give a person access to the machine.
- There are several attributes associated with each account on a system.

Attributes of a User Account

- Username
- Password (password placeholder)
- User ID (UID)
- Primary Group ID (GID)
- Textual Description (Gecos)
- Home Directory
- Default Shell

What is a Group

- A group is a set of users that allows permissions to be granted in a more organized way.
 - File access
 - Used directly by some programs like sudo

Group Attributes

- Group Name
- Group ID (GID)
- Group Members

Special accounts

- System processes (daemons) need accounts on the machine too and you will sometimes see accounts for specific programs
 - Limit elevated access as much as possible
- What is running on the system

```
ps -eo user|sort|uniq -c
```

```
ps -eo user,pid,comm | sort
```

In the beginning

- User information stored in files
 - /etc/passwd
 - /etc/group
- Password encrypted but vulnerable to attack
- All credentials local

Hiding secrets

- /etc/passwd
 - User name, UID, etc
- /etc/shadow
 - Encrypted password, password aging

/etc/passwd format

mscott:x:5001:5000:Michael Scott:/home/mscott:/bin/bash

mscott - user name

x - password is encrypted and stored in /etc/shadow

5001 - User ID

5000 - Group Id

Michael Scott - Full name (GECOS)

/home/mscott - home directory

/bin/bash - login shell

/etc/shadow format

mscott:\$6\$hyz\$bx0cpZOlf0:18056:0:99999:7::

mscott - user name

\$6\$hyz.... encrypted password

- \$x\$ sets encryption algorithm
- First few digits is the salt

18056 - Last date password changed

0 - Minimum password age

99999 - Maximum days password is valid

7 - Days of warning before expiration

Inactive and Expire is blank

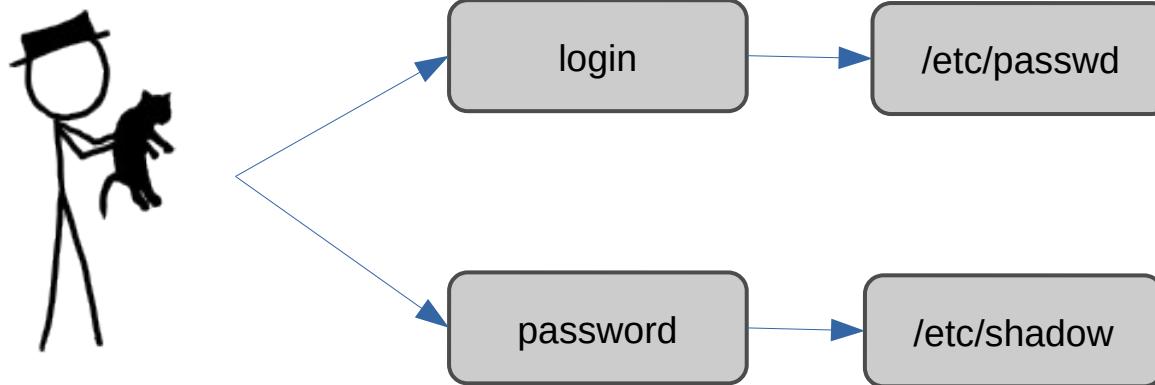
/etc/group format

managers:x:6001:mscott,jhalpert,dschrute

- managers - group name
- x - password in /etc/gshadow
- 6001 - Group ID
- mscott, jhalpert, dschrute - members
 - Users with the GID as their primary group will also be members of this group

Basic login

- User credentials checked against files



Networked User Information

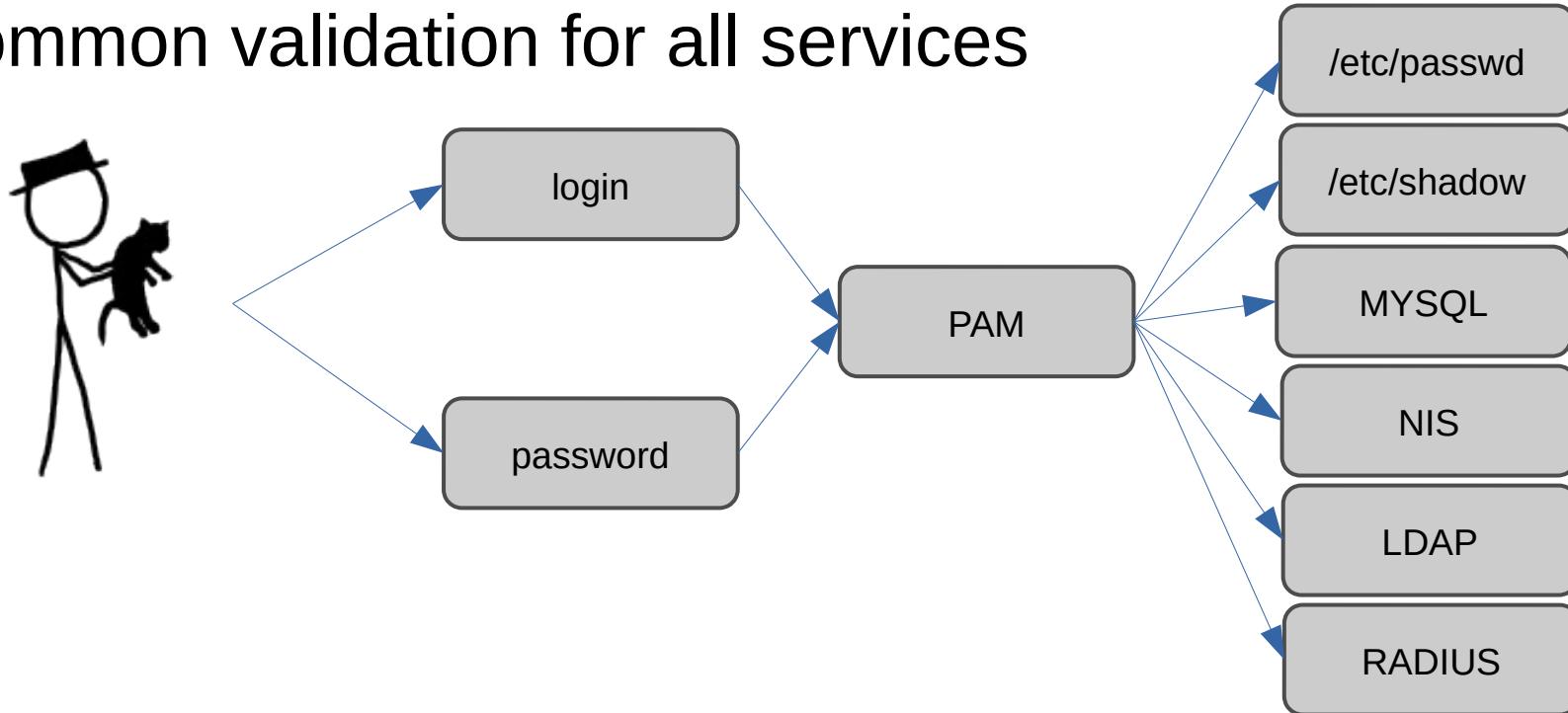
- NIS - Network Information Service
 - Originally called Yellow Pages
- LDAP - Lightweight Directory Access Protocol
- Kerberos
- many others

Which login is used?

- NSS - Name Service Switch
 - Sets order of evaluation for GNU C library
- PAM - Pluggable Authentication Module
 - Fine grained control of authentication

PAM login

- Pluggable Authentication Modules
- Common validation for all services



Adding a user

- `vipw`
 - Trusted vi /etc/passwd
 - Also -p
 - `-s` to edit /etc/shadow
 - `-g` to edit /etc/group
 - Also `vigr`
- `useradd login`
 - `u` - set userid
 - `U` - add private group
 - `g` - set GID or group
 - `G` - add groups
- Must set password
- Template for files in home directory in /etc/skel

passwd

- Allows the user to update their own password
- `passwd login`
 - Allows root to set the password for a user
- `chpasswd`
 - Change password in batch mode

groupadd

- Creates a new group
- -g sets GID
- -p can set a group password
- `groupadd -g 6001 managers`

adduser

- debian
 - Like useradd, but also prompts for name, GECOS and password
 - `adduser login group`
 - Adds *login* to *group*
- RedHat
 - Basically just useradd
- BSD
 - Detailed prompts

Removing a user and group

- `userdel login`
 - Removes the user
 - `-r` also remove home directory and mail spool
- `groupdel group`
 - Removes the group

`usermod` *login*

- Modifies the user attributes
 - l change login name
 - u change UID (also modifies mailbox and files in /home)
 - L lock (disable) login by removing password
 - g set primary group for user
 - G add user to this secondary group
 - p change password
 - s change shell

Change user info and shell.

Users can also change their own

- chfn
 - Change “finger” data
 - Full Name
 - GECOS data
 - Office
 - Phone number
- chsh
 - Change login shell for user
 - -l list available shells

Special groups

- wheel - traditional users with root access
 - Redhat: users with unlimited sudo
 - debian: wheel group renamed to sudo
- dialout - users with access to serial ports
- Private groups

Gotchas

- Group membership is updated on login
 - Log out or launch new login shell

File Permissions

Linux System Administration
Fall 2023

UNIX File Modes

- A unix file's mode is a collection of numbers that represents permissions.
 - Fundamental controls user's access to system.
 - r - Read controls data that can be read
 - w - Write controls what data can be written
 - x - Execute controls what commands can be run

Unix file modes

Permission	Symbolic	Binary	Octal
everything	rwx	111	7
read and write	rw-	110	6
read and execute	r-x	101	5
read only	r--	100	4
write and execute	-wx	011	3
write only	-w-	010	2
execute only	--x	001	1
none	---	000	0

Access Modes

- File
 - **r** can read file
 - **w** can modify file
 - **x** can execute file
 - run file as command
- Directories
 - **r** can see file names
 - **w** can add or remove files
 - **x** allow access to files in the directory
 - access to inodes

File mode

- Special:User:Group:Other
 - Special - SetUID,SetGID,Sticky
 - User - File owner access
 - Group - Group access
 - Other - Access by all others

Special Bits

- SetUID
 - Run executable with UID of owner
- SetGID
 - Executable: Run with GID of owner
 - Directory: New files inherit directory group
- Stricky bit
 - Legacy: Keep executable in swap on exit
 - Directory: Only root or owner can delete or rename files
 - Files: Usually ignored

r and x

- Binary executables only need x
 - Executed directly by the kernel
- Shell scripts need r and x
 - Interpreter needs to read the file
- Directories
 - r required to list files
 - x required to access files

File mode examples

-rw-----	1	mscott	mscott	/home/mscott/.ssh/id_rsa
-rw-r--r--	1	root	root	/etc/issue
-rwxr-xr-x	1	root	root	/usr/bin/zip
-rwsr-xr-x	1	root	root	/usr/bin/passwd
-rwxr-sr-x	1	root	tty	/usr/bin/wall
drwxrws---	1	root	sales	/home/sales
drwxrwxrwt	25	root	root	/tmp

Manipulating owner & permissions

- `ls -l` show permissions
- `stat` show detailed file permissions
- `chmod` sets permissions (-R for recursive)
- `chown` change user[:group] (-R for recursive)
- `chgrp` change group (-R for recursive)

chmod symbolic

- [ugoa] [+-=] [rwxst]
 - **u**ser, **g**roup, **o**ther, **a**ll
 - + add, - remove, = set
 - **r**ead, **w**rite, **e**xecute, **s**et^{*}**i**d, **s**ticky
- Examples
 - u+w add write permission for user
 - a-x remove execute permission for all
 - ug=rw,o=r set user & group to rw, others to r

chmod numeric

644	r <u>w</u> -r--r--	u=rw,g,o=r
755	r <u>wx</u> r-xr-x	u=rwx,g,o=rx
0755	r <u>wx</u> r-xr-x	u=rwx,g,o=rx
4755	rws <u>r</u> -xr-x	u=rwxs,g,o=rx
2775	rwx <u>rwsr</u> -x	u=rwx,g=rwxs,o=rx
1777	rwx <u>rwxrwt</u>	ug=rwx,o=rwxt

umask

- Sets file creation mode mask
 - In bash this is a shell builtin command
 - Set on login
- umask sets permissions to mask
 - If the bit is set in umask, it is 0 in the file mode
 - Result is perm & ~umask
- umask is often 0002 (o-w) or 0022 (go-w)

Group access

- Group access is critical to several labs
 - Allow users to share resources
 - Read access to files
 - Write access to files
- Groups define users with similar privileges
 - Shared access
- SetGID and umask determine defaults

Access Control Lists (ACL)

- Set permissions for individual users
 - `setfacl -m mscott:rwx /home/accounting`
- Show ACL
 - `getfacl /home/accounting`
 - A trailing + in `ls -l` indicates presence of ACL
- Not all filesystems support ACLs
 - ACL support can be set by mount
 - Should be a very last resort
- **If you resort to ACL in the labs, you are doing it wrong!!!**

setcap/getcap

- **capabilities** are privileges normally reserved for root that can be selectively granted to executables
 - `setcap cap_net_raw=ep command`
command can access raw sockets without being root
 - `getcap command`
show capabilities associated with command
- More selective than setuid

• Example capabilities

man capabilities show all capabilities

- CAP_CHOWN
 - Change file ownership
- CAP_KILL
 - Send any signal
- CAP_NET_RAW
 - Access to raw sockets
- CAP_SYS_RAWIO
 - Acces to hardware
- CAP_SYS_TIME
 - Access to real time clock
- CAP_SYSLOG
 - Access to logging

Root Access

Linux System Administration
Fall 2023

Being root

- There are no restrictions on what root can do
- When you are root, a typo can brick the system
- If root login is compromised, access is complete

Playing it safe

- Limit access to root login
 - Disable account or limit account to local logins
- Grant root access as needed with sudo
 - Fine grained control on commands
- Grant access via trusted commands
 - Special permission bits
 - Special file capabilities

Permissions Reserved for root

- Mounting a filesystem
- Changing a file or folder user & group ownership
- Bypassing the kernel to obtain or send network traffic on the physical wire itself
- Listening on TCP / UDP ports lower than 1024
- Controlling systemd
 - Reboot, start/stop system processes, etc.

Limiting root logins

- Set root login shell to /sbin/nologin
- PAM
 - /etc/pam.d/system-auth
 - /etc/pam.d/sshd
 - /etc/security/access.conf
- /etc/ssh/sshd_config
 - PermitRootLogin
 - no
 - yes
 - prohibit-password
 - forced-commands-only

root login policy

- No root access is the most restrictive
 - Could be a problem if the machine is hung
- No ssh logins is good for DMZ or untrusted net
 - Console access can be a useful backup
- Limited ssh logins
 - Good for remote administration

sudo



- su "do" (super user do)
- Preferred way of elevating permissions
 - Allow by user and by system command access
 - Logging of commands
- Periodically require password
- Partly developed at CU Boulder

sudo configuration

- Main configuration
 - /etc/sudoers
- Local configuration
 - /etc/sudoers.d/*
 - File name does not matter
 - Choose by function
 - Facilitates distributed management
- Edit configuration files using visudo

sudo environment

- env_reset
 - Limits variables from user shell transferred to root
- secure_path
 - Modifies PATH used to find commands
- Integrates with PAM

Permitted Commands

- Allow root and wheel (or sudo) to do anything
 - root ALL=(ALL) ALL
 - %wheel ALL=(ALL) ALL
- Generic format
 - user host = command
 - user host = (user) command
 - % refers to a group
 - , for a list
 - ! to exclude
 - ALL means no restriction

Aliases

- **Host_Alias**

```
Host_Alias FILESERVERS = www, nfs
```

- **User_Alias**

```
User_Alias MANAGERS = mscott, jhalpert
```

- **Cmnd_Alias**

```
Cmnd_Alias START /usr/bin/systemctl start,/usr/bin/systemctl restart
```

- **Aliases are context sensitive**

Wildcards

- Match command string
 - Could be the command, parameter or files
 - Symbol matching, e.g. [0–9]
 - Wildcards
 - * zero or more characters
 - ? zero or one characters
- Command allowed if it matches any line in sudoers
- Matches are BNF not regexes
 - Exclude matches with !

Wildcard examples

- Allow mscott access to /var/log/messages and backups
 - mscott ALL=/bin/cat /var/log/messages
 - mscott ALL=/bin/cat /var/log/messages.[1-9]
- Never end a command in *
 - Allows users to sneak additional stuff at the end of the command
 - mscott ALL=/bin/cat /var/log/messages*
allows
 - sudo cat /var/log/messages.1 and
 - sudo cat /var/log/messages /etc/shadow

No password

- Execute a command with elevated privileges without requiring the usual sudo password

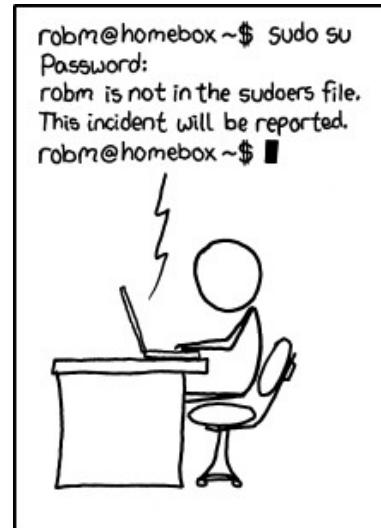
```
mscott ALL=NOPASSWD: /usr/bin/systemctl restart apache2
```

- ***Use with caution!!!***

- Password encourages the user to think before acting
 - Limit the command to the absolute minimum

sudo logging

- Commands run using sudo are logged
 - debian /var/log/auth.log
 - redhat /var/log/audit/audit.log
- Remote logging possible



sudo tips

- Test your configurations thoroughly
 - Could be a significant security hole
- Use aliases to simplify user experience
- Maintain customization in /etc/sudoers.d/
 - Works best with distributed management

Executables with privilege escalation

- setuid
 - run using UID of file owner of executable
- setgid
 - runs using GID of file group of executable
- setcap
 - Allow executable access to specific facilities otherwise requiring root

Extra Permission Bits

NUMBER	PERMISSION	RWX
7	setuid, setgid, stickybit	111
6	setuid and setgid	110
5	setuid, and stickybit	101
4	setuid	100
3	setgid, and stickybit	011
2	setgid	010
1	stickybit	001
0	none	000

Changing your password

- **/etc/shadow**
 - Ownership root : shadow
 - Permissions -rw-r-----
- **/usr/bin/passwd**
 - Ownership root : root
 - Permissions -rwsr-xr-x
 - Program is trusted to change the right user

Secure Shell

Linux System Administration
Fall 2023

Remote System Access

- The goal
 - Execute commands on a remote system
 - Remote shell or single command
 - Transfer files between systems
- Requirements
 - Invisible - terminal window just like local
 - Secure - the network cannot be trusted

In the beginning

- telnet (**teletype network**)
 - Original TCP/IP based remote terminal (RFC 15)
 - Plain text with lightweight escape sequences (0xff=IAC)
 - Port 23
 - Unsafe for untrusted networks
 - Still useful in testing TCP/IP connections
 - Sometimes enabled in legacy hardware
- Replaced by **secure shell (ssh)**

BSD r-commands

- Added in BSD 4.1 (1981)
 - Often remapped to ssh, scp, etc on many systems

Client	Function	Daemon	Port	Protocol
rexec	command	rexecd	512	TCP
rlogin	shell	rlogind	513	TCP
rsh	command	rshd	514	TCP
rcp	copy			
rstat	stat	rstatd		UDP
ruptime	uptime	rwhod	513	UDP
rwho	who			

Secure Shell (SSH)

- Designed by Tatu Ylönen 1995
 - OpenSSH - Implementation by OpenBSD (1999)
 - dropbear - Tiny footprint ssh for embedded devices
- Secure login to remote system
 - Public-key encryption over TCP port 22
 - Basis for scp, sftp, etc
 - Supports tunneling (X11, port forwarding)
 - Allows password-less logins

The importance of ssh

- Goal of exams are in large part establishing an ssh connection to a remote server
 - Basic machine functions are up
- Tunneling allow access to network services as if they are local

Using ssh

- ssh user@host
 - p port (default 22)
 - v verbose (-vv or -vvv for even more verbose)
 - L local:host:remote (tunnel)
 - R remote:host:local (reverse tunnel)
 - X Forward X11 (-Y for secure)

ssh examples

```
ssh www.dm.com
```

```
ssh -p 2222 root@broken.dm.com
```

```
ssh -v root@unreachable.dm.com
```

```
ssh -L 2222:10.21.32.2:22 machinea
```

```
ssh -p 2222 localhost
```

```
scp -Cp -P 2222 localhost:foo .
```

ssh files

- `/etc/ssh/ssh_config`
 - Global configuration
 - Can be by host or *
 - Override options with `ssh -o`
- `/etc/ssh/sshd_config`
 - Incoming connections
- `~/.ssh/`
 - `known_hosts`
 - `signature`
 - `authorized_keys`
 - public key(s)
 - `id_rsa`, `id_ed25519`, ...
 - private key

Copying files with scp

- `scp localfiles host:remotedir`
`scp host:remotefiles localdir`
 - p preserve file times and modes
 - r recursive
 - P Port (default 22)
 - C Compress

Generating keys

- Generate key pair
 - ssh-keygen
 - t key type (rsa, rsa-sha2-512, dsa, ecdsa, ed25519, etc.)
 - b key length (e.g. 4096)
 - C comment
 - Default output to ~/.ssh/
 - Empty passphrase is OK
- ssh-keygen -t rsa -b 4096 -C saclass@colorado.edu
 - Private key ~/.ssh/id_rsa
 - Public key ~/.ssh/id_rsa.pub

Distributing keys

- By hand

```
scp ~/.ssh/id_rsa.pub remotehost:  
ssh remotehost  
cat id_rsa.pub >> ~/.ssh/authorized_keys  
rm id_rsa.pub
```

- `ssh-copy-id remotehost`

Using ssh in scripts

- ssh will only accept a password interactively
 - Requires password-less authentication
- sshpass allows ssh to be run non-interactively but with a password
 - PW=*secret*
`sshpass -p $PW ssh host command`
 - Could be a security hole

Notes about key pairs

- The local key must be in \$HOME/.ssh/
 - What is \$HOME when you do sudo?
- The remote key must be in \$HOME/.ssh/ for the remote user you are login in as

Pause for Paranoia

- Password-less logins extends the domain of trust to machines with the private key
 - Allows tunnels through firewalls
 - DMZ hosts should never have private keys for the LAN
 - Set up access from the LAN side
 - root password-less access should be minimized
 - Agonize over laptops

rsync

- Fast tool for copying files
 - Modern versions transparently run over ssh
 - Conditionally copies files based on time and size
 - Clever block checksum for updating files
 - Can preserve permissions, links and holey files
- Best tool to mirror directory trees

rsync options

- | | |
|----------------------------|------------------------------------|
| -a archive | -S write holey files |
| -p preserve permissions | -n dry run |
| -t preserve times | -v verbose |
| -r recursive | -z compress |
| -l preserve symbolic links | --delete-excluded |
| -o preserve owner | --existing (no new files) |
| -g preserve group | --ignore-existing (only new files) |
| -H preserve hard links | --exclude= <i>pattern</i> |
| -A preserve ACLs | --bwlimit= |

rsync tips

- Watch out for a trailing / on the source
- Use -n to do a dry run before big transfers
 - Especially with --delete-excluded
- *rsync* can be used locally as a clever *cp*

Copy trees with tar

```
ssh user@host tar cf - -C dir | tar xf -
```

- **ssh** to *host* and run **tar**

- C** does a chdir to *dir*

- cf** - creates and archive to stdout (-)

- **tar xf** -

- unpack archive from stdin (-)

- archive is piped over ssh

Important *sshd_config* options

- PermitRootLogin
 - yes = allowed
 - prohibit-password = must use key pair
 - forced-commands-only = key pair only, no shell
 - no = never
- UseDNS (should be yes)
- UsePAM (should be yes)

mosh

- mosh is an alternative to ssh that works better over unreliable connections
 - ssh takes a long time to recover from lost packets
 - mosh uses UDP
 - mosh tolerates changing the source IP address
- Good choice for mobile (cellular)
 - Example implementation: blink for iOS

Package Management

Linux System Administration
Fall 2023

In the beginning...

- Software on CD
- **autoconf** to deal with OS & compiler peculiarities
- Build from source

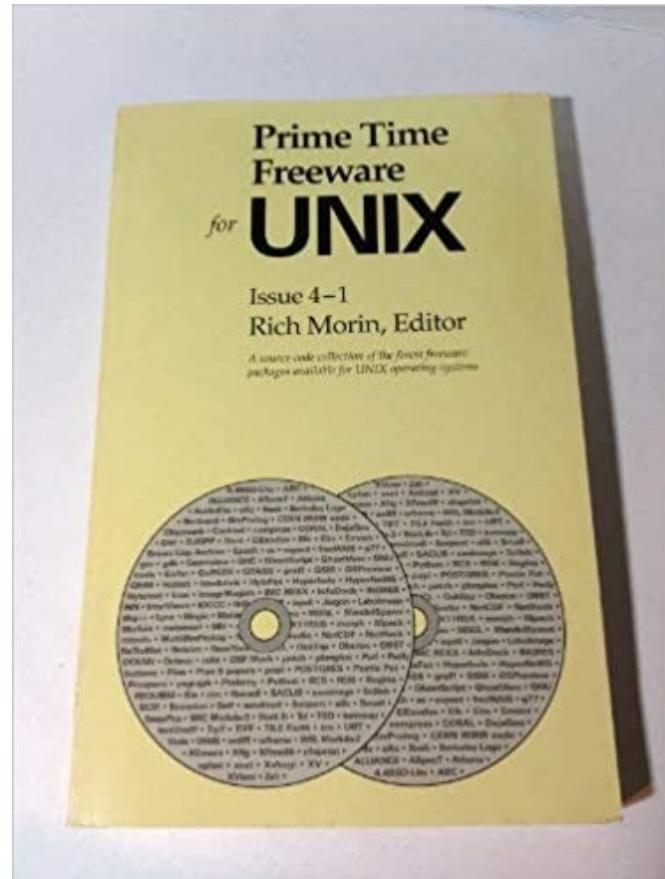
```
tar xzf apache_1.3.6.tar.gz
```

```
cd apache_1.3.6
```

```
./configure --prefix=/usr
```

```
make
```

```
sudo make install
```



Then came package managers

- Pre-compiled binaries and configuration files
- Checks shared library dependencies
- Specialized archive to include config scripts
 - Redhat rpm (.rpm)
 - Debian dpkg (.deb)
- CD or download



Package manager systems

- With the internet came repositories
 - Download on demand
- Package management systems download the package as well as its dependencies
 - Red Hat **yum** (now called **dnf**)
 - Debian **apt-get** (also **apt**)
 - Arch **pacman**
 - OSX **homebrew**
 - BSD **pkg**

Advantages

- Simplifies downloading and installing packages
- Facilitates management of dependencies
- Automate the process of updating packages
- Supports cleanly removing packages
 - but not necessarily dependencies

Package Repositories

- Collection of packages for a distribution
- Accessed via HTTP/HTTPS
- Mirrors spreads bandwidth load
 - Update mirrors with rsync
 - Many sites run a local mirror
- Repositories may have multiple distros and versions

RedHat EPEL

- Extra Packages for Enterprize Linux
 - Additional packages not included in RHEL
 - Adds more bleeding edge packages from Fedora
- **dnf install epel-release**
 - Add EPEL as an additional repository

Repository Configuration

- RedHat

/etc/yum.repos.d/*

- dnf still uses yum.repos.d

Rocky

/etc/yum.repos.d/rocky.repo

EPEL

/etc/yum.repos.d/epel.repo

- debian

/etc/apt/sources.list

deb *URL* release main

Package naming

- RedHat
 - package-version.os.arch.rpm
 - openssh-8.7p1-8.el9.x86_64.rpm
 - perl-base-2.27-479.el9.noarch.rpm
- debian
 - package_version+os_arch.deb
 - openssh-client_8.4p1-5+deb11u1_amd64.deb

Making sure all packages are current

- RedHat
 - dnf update
 - check metadata
 - check for updates
 - Confirmation defaults to NO
- debian
 - apt-get update
 - check metadata
 - apt-get upgrade
 - check for updates
 - Confirmation defaults to YES

Installing dig

- Redhat

dnf provides dig

- bind-utils

dnf install bind-utils

- Dependencies installed

bind-libs

bind-license

fstrm

libmaxminddb

libuv

protobuf-c

- debian

apt-file search /usr/bin/dig

- bind9-dnsutils

apt-get install bind9-dnsutils

- May already be installed

Installing a downloaded package

- rpm -i foo.rpm
- dpkg -i foo.deb
 - Could fail and show dependencies not met
- dnf install ./foo.rpm
- apt-get install ./foo.deb
 - Will fetch dependencies as needed

Finding out about packages

- Redhat
 - Find string in package name or description
 - `dnf search string`
 - Package metadata
 - `dnf info package`
- debian
 - Find string in package name or decription
 - `apt search string`
 - Package metadata
 - `apt show package`

What package installed dig?

- RedHat

```
rpm -qf /usr/bin/dig  
bind-utils-9.16.23-1.el9.x86_64
```

- debian

```
dpkg -S /usr/bin/dig  
bind9-dnsutils: /usr/bin/dig
```

What files were installed by bind-utils?

- RedHat

```
$ rpm -ql bind-utils  
...  
/usr/bin/arpaname  
/usr/bin/delv  
/usr/bin/dig  
/usr/bin/dnstap-read  
/usr/bin/host  
/usr/bin/nslookup  
/usr/bin/nsupdate  
...
```

- debian

```
$ dpkg-query -L bind9-dnsutils  
...  
/usr/bin/delv  
/usr/bin/dig  
/usr/bin/dnstap-read  
/usr/bin/mdig  
/usr/bin/nslookup  
/usr/bin/nsupdate  
...
```

What packages are installed

- RedHat
 - rpm -qa
- debian
 - dpkg -l
 - ii installed
 - un uninstalled

Removing packages

- Redhat
 - dnf remove *package*
 - rpm -e *package*
 - dnf autoremove
 - Removes obsolete
- debian
 - apt-get remove *package*
 - --purge removes configs
 - dpkg -r *package*
 - dpkg -P to purge
 - apt autoremove
 - Removes obsolete

Configuration files in packages

- Packages may contain configuration files
 - What happens if a new release changes the configuration file, but you changed the config?
- Redhat
 - Saves new file with .rpmsave or .rpmnew appended
- debian
 - Prompts to keep, replace, modify, etc.
- Generally you will need to re-apply your edits

Some sage advice

- When changing a configuration file, save a copy of the original
 - cp -a foo.conf foo.conf.0
 - diff shows changes
 - Helps to recover from failures
 - Reboot if appropriate
- RedHat does not enable packages automatically

Installing groups of packages

- Redhat
 - dnf groupinstall “Development Tools”
- debian
 - apt install build-essential
- Installs package collections
 - gcc, g++, make, binutils, header files, and lots more

Automated installation

- Bare metal installation from a script
 - Used in data centers or compute clusters
- RedHat/Anaconda kickstart
- debian FAI (Fully Automatic Installation)

Apache

Linux System Administration
Fall 2023

What is Apache?

- One of the original web servers
 - Initially released in 1998
- Stable, very configurable and widely used
- Part of LAMP
 - Linux
 - Apache
 - Mysql
 - PHP/Perl/Python

Alternatives to Apache

- nginx
 - more geared to serving static content
- lighttpd
 - good security and performance
- many others

Quick Start

- Redhat
 - dnf install httpd
 - Enable and start httpd
 - systemctl enable httpd
 - systemctl start httpd
 - Disable and stop firewall
 - systemctl disable firewalld
 - systemctl stop firewalld
 - or punch a hole in the firewall
 - firewall-cmd --zone=public --add-port=80/tcp --permanent
 - firewall-cmd --reload
- Debian
 - apt install apache2
 - Enabled and started by default
 - No firewall
- On your Virtualbox
 - Forward (say) 800x to port 80

Connect with a browser

- RedHat

HTTP Server Test Page

This page is used to test the proper operation of an HTTP server after it has been installed on a Rocky Linux system. If you can read this page, it means that the software is working correctly.

Just visiting?

This website you are visiting is either experiencing problems or could be going through maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you've expected, you should send them an email. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

The most common email address to send to is:
"webmaster@example.com"

Note:

The Rocky Linux distribution is a stable and reproducible platform based on the sources of Red Hat Enterprise Linux (RHEL). With this in mind, please understand that:

POWERED BY  POWERED BY 

- debian

Apache2 Debian Default Page

 It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
-- mods-enabled
|   '-- *.load
|       '-- *.conf
-- conf-enabled
|   '-- *.conf
-- sites-enabled
    '-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

HTTP or HTTPS?

- HTTP
 - Insecure
 - Port 80
 - Plain TCP/IP connection
- HTTPS
 - Secure
 - Port 443
 - TLS (was SSL)
 - Transport Layer Security
 - Requires PKI certs

Where do I find things?

- Redhat
 - Configuration
 - /etc/httpd
 - HTML files
 - /var/www/html
 - CGI files
 - /var/www/cgi-bin
 - Logs
 - /var/log/httpd
- debian
 - Configuration
 - /etc/apache2
 - HTML files
 - /var/www/html
 - CGI files
 - /usr/lib/cgi-bin
 - Logs
 - /var/log/apache2

Apache Security

- Web services are a common target
- The server runs as a non-priviliged user
 - Redhat apache
 - Debian www-data
- Directories must be readable by user or group
 - CGI programs sometimes need write access

Apache Concepts

- Static files are in the DocumentRoot
 - File are served verbatim
- Dynamic content is the result of running a program (CGI)
 - Common Gateway Interface
 - Request on `stdin`
 - Response on `stdout`
 - `stderr` to error log
- Modules to perform varous actions
 - Access control
 - Rewriting URLs
 - Proxy and load balancing
- Named and IP based Virtual web servers
 - One server, many sites
 - Highly parallel

Basic configuration

- Configuration files looks like HTML
 - <Directory "/var/www/cgi-bin">
 - <VirtualHost "foo.bar.com:80">
- RedHat puts everything in httpd.conf
 - Does include a files in /etc/httpd/conf.d
- debian spreads configuration over multiple files and directories
 - Provides a2en* a2dis* commands to modify
 - apachectl
 - -S shows content
 - -t checks syntax

Important Keywords

- ServerName FQDN for this host
- ServerRoot Root directory root for configuration
- Listen IP and port to bind to
- DocumentRoot Root directory for files
- SSLEngine Enable SSL/TLS
- SSL*Cert* Certificates

Directory Directives

- <Directory *path*> select directory
- Options **control** apache behavior for this path
 - Indexes **allow** directory listing
 - FollowSymLinks **follow** symbolic links
 - ExecCGI **allow** CGI
- AllowOverride **controls** .htaccess
- Require **set** authorization restrictions

HTTP Exchange

Request sent to TCP port 80

GET / HTTP/1.1

Host: dundermifflin.com

User-Agent: Mozilla/5.0

Accept: text/html;

Connection: keep-alive

Response

HTTP/1.1 200 OK

Date: Mon, 23 May 2005 22:38:34 GMT

Content-Type: text/html; charset=UTF-8

Content-Length: 155

Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT

Server: Apache/1.3.3.7 (Red-Hat/Linux)

Connection: close

<html>.....</html>

Adding a virtual Host

```
<VirtualHost *:80> IP adddress (*) and port  
  ServerName    FQDN of virtual host  
  ServerAlias   Alternative FQDN of virtual host  
  ServerAdmin   email of admin  
  DocumentRoot  directory containing HTML files  
  ErrorLog      error log file  
  CustomLog     access log file  
</VirtualHost>
```

Rewriting Rules

- Permanent remap (HTTP to HTTPS)
 - Redirect permanent / https://dm.com/
- Replace URL
 - RewriteEngine on
 - RewriteRule ^/?old([a-zA-Z.]*)\$ /new\$1 [R=301,L]
 - Changes oldxxxxx to newxxxxx
- Requires mod_rewrite

icons

- Images are stored in /usr/share
 - Debian /usr/share/apache2/icons
 - RedHat /usr/share/httpd/icons
- Used in index and HTML files
- Varies between machines

CGI scripts

Python

```
#!/usr/bin/python

import os;
print("Content-type: text/html\r\n\r\n");
print("<H2>Environment</H2>\r\n");

for param in os.environ.keys():
    print("%s=%s<br>\n" % (param,os.environ[param]));
```

Perl

```
#!/usr/bin/perl

print "Content-type: text/html\r\n\r\n";
print "<H2>Environment</H2>\r\n";

while (my ($key,$val) = each %ENV)
{
    print "$key=$val<br>\n";
```

Hardening Apache

- fail2ban
 - blocks IP using iptables based on logs
- selinux (RedHat)
- apparmor (Debian)
 - limits Apache access to certain directories

Scheduling Tasks

Linux System Administration
Fall 2023

Scheduling Commands

- cron
 - Run a recurring command
- at
 - Run a command at a specified time
- systemd timers
 - Re-invents the wheel

Common periodic tasks

- Rotate logs
- Check for software updates
- Perform backups

cron files

- One liners
 - /etc/crontab system-wide entries
 - /etc/cron.d additional crontab entries
- Scripts
 - /etc/cron.hourly script that run hourly
 - /etc/cron.daily scripts that run daily
 - /etc/cron.weekly scripts that run weekly

crontab entries

min hour dom mon weekday user command

- * = don't care
- */int = match every (*/10 = 0, 10, 20, 30, 40, 50)
- integer = match exactly
- int-int = range (1-5 = 1, 2, 3, 4, 5)
- int,int,int = match from list

Sunday is weekday 0

crontab examples

- At 00:15 each day, rsync backup from machinea

```
15 0 * * * root rsync -aH machinea:/backup/* /backup
```

- Run the checkwx script on the hour

```
0 * * * * root /usr/local/bin/checkwx
```

- Run the ntsgw script every 5 minutes

```
*/5 * * * * willem /usr/localbin/ntsgw
```

- Run the backup-db script Sunday morning at 2am

```
0 2 * * 0 root /usr/local/bin/backup-db
```

- Run the snapshot script at 00:30 on the first of the month

```
30 0 1 * * root /usr/local/bin/snapshot
```

at

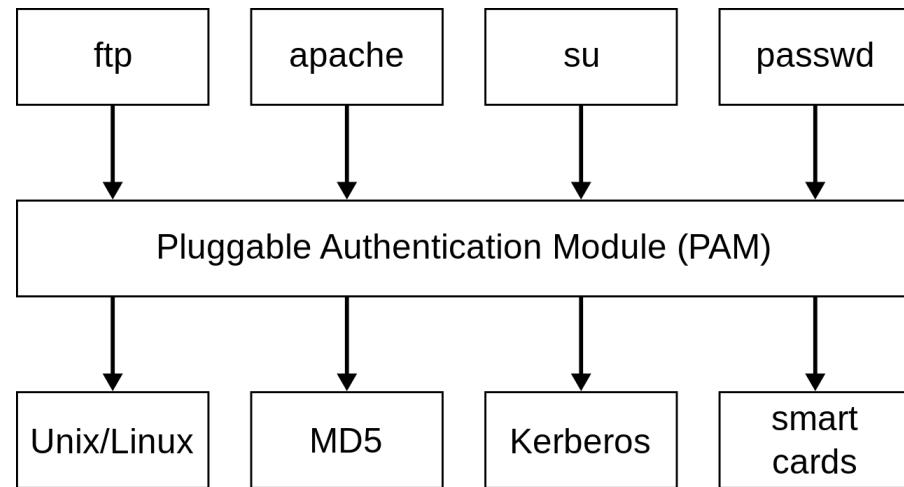
- Commands to run are read from the command line and executed as /bin/sh
 - at 21:00 at 9pm
 - at now+20 minutes (20 minutes from now)
- atq show jobs queued
- atrm remove job from queueq

Pluggable Authentication Modules (PAM)

Linux System Administration
Fall 2023

What is PAM?

- Common authentication module used by many services to authenticate users
 - Allows fine grained control for user access
 - Single service to simplify authentication



Important Password Attributes

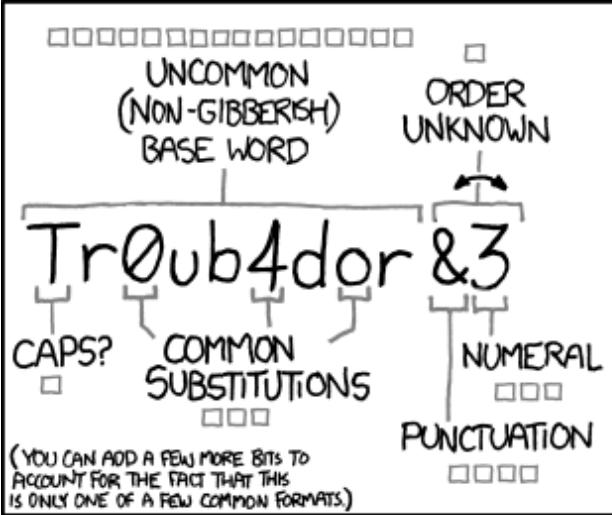
- Username
- Password hash (Method + Salt + Hash)
- Last date changed
- Maximum age
 - Warning time
 - Inactive time
 - Expire

Password Threats

- Hashes from /etc/shadow or elsewhere
- Plain-text passwords in scripts
- Sniffing of insecure protocols
- Shoulder surfing
- Keyboard loggers (hardware or software)
- Brute force guessing or dictionary attacks

Mitigating password threats

- Exclude nonessential users from the machine
- Enforce passwords that are randomly generated, 12 characters or longer and from a set of 94 characters, symbols and numbers
- Using a key fob or application to generate key
- Two factor authentication
- Biometrics



~28 BITS OF ENTROPY

$2^{28} = 3$ DAYS AT 1000 GUESSES/SEC

(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)

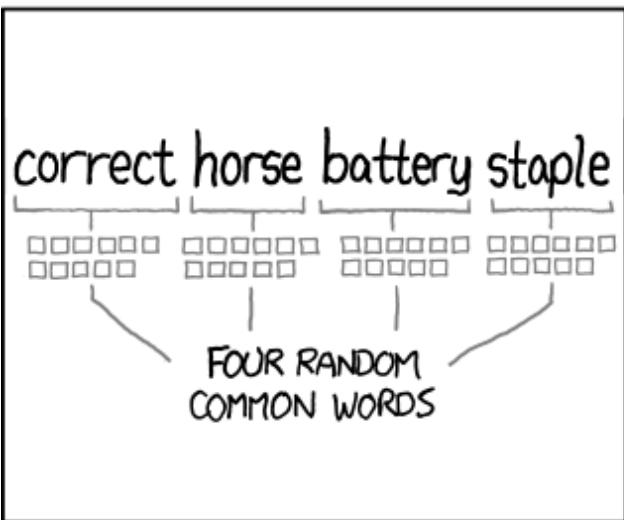
DIFFICULTY TO GUESS: EASY

Detailed description: This panel shows a password with approximately 28 bits of entropy. It includes a calculation ($2^{28} = 3$ DAYS AT 1000 GUESSES/SEC) and a note about a plausible attack on a weak remote web service.

WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE 0s WAS A ZERO?
AND THERE WAS SOME SYMBOL...

DIFFICULTY TO REMEMBER: HARD

Detailed description: This panel features a stick figure thinking about a password. The thought bubble contains the text 'WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE 0s WAS A ZERO?' and 'AND THERE WAS SOME SYMBOL...'. Below the thought bubble, it says 'DIFFICULTY TO REMEMBER: HARD'.



~44 BITS OF ENTROPY

$2^{44} = 550$ YEARS AT 1000 GUESSES/SEC

DIFFICULTY TO GUESS: HARD

Detailed description: This panel shows a password with approximately 44 bits of entropy. It includes a calculation ($2^{44} = 550$ YEARS AT 1000 GUESSES/SEC) and a note that it's 'HARD' to guess.

THAT'S A BATTERY STAPLE. CORRECT!

DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT

Detailed description: This panel features a stick figure thinking about a password. A thought bubble contains the text 'THAT'S A BATTERY STAPLE.' and 'CORRECT!'. Below the thought bubble, it says 'DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT'.

THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Common sense password policy

- Enforce passwords that are strong enough, and educate them on selecting a good memorable password and force them to change it periodically.
- Use two factor authentication for critical systems

Password Expiration Example

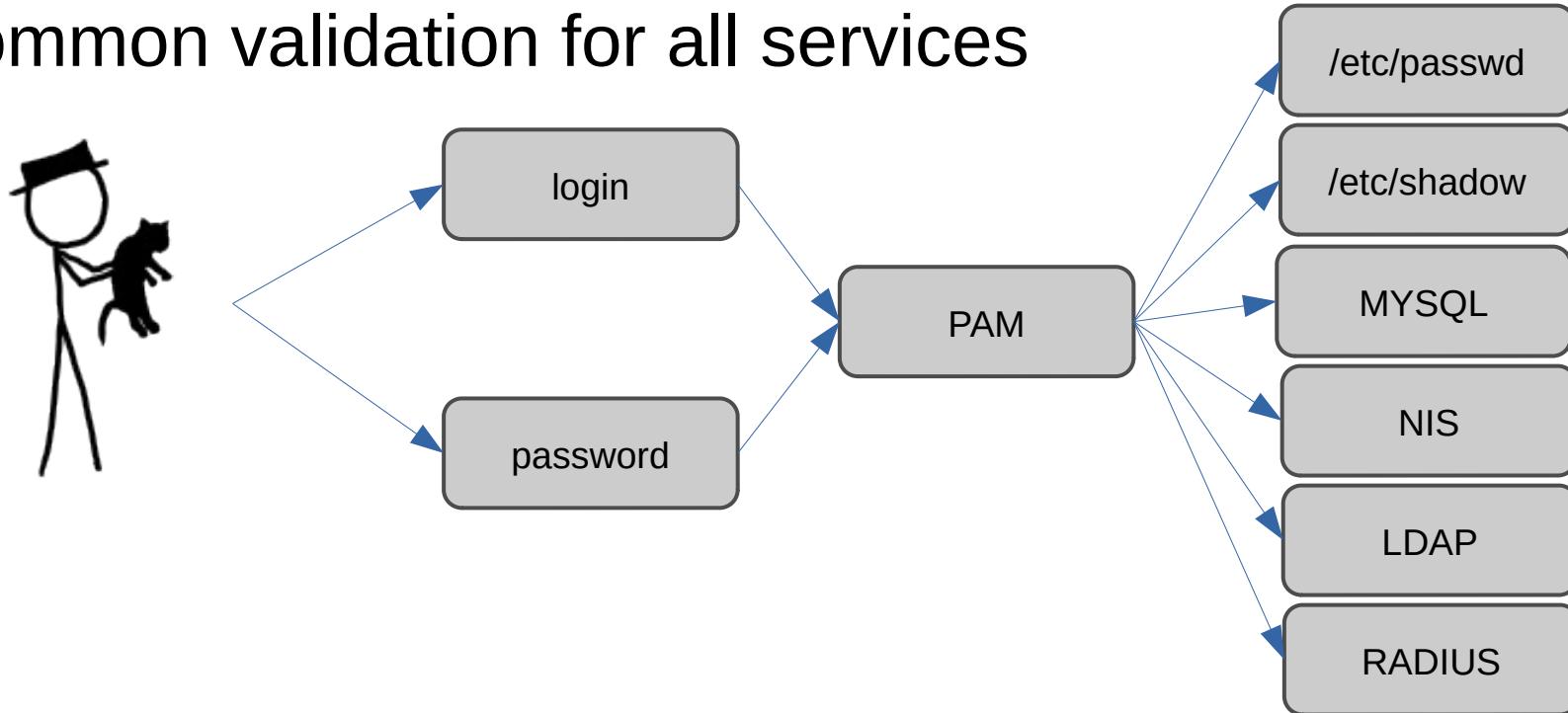
- mscott:\$6\$88afba...:19245:0:90:10:7::
 - 19245 = Sep 10, 2022
 - 90 = Maximum age (Dec 9, 2022)
 - 10 = Warning period (Nov 29, 2022)
 - 7 = Inactivity period (Dec 15, 2022)
 - account expiration period (never)

Setting password aging

- `chage user`
 - l show current settings
 - d 0 force change at next login
 - E set expiration date
 - l lock account
 - u unlock account

PAM login

- Pluggable Authentication Modules
- Common validation for all services



PAM Modules

- Authorization
 - Identifies users, groups and membership
- Account
 - Enforces login restrictions
- Session
 - Takes cares of tasks like creating or mounting home directory
- Password
 - Deals with password changes

PAM Files

- /etc/pam.d
 - PAM configuration files
- /etc/security
 - Policy files
 - access.conf - who can login from where
 - pwquality.conf - password requirements

Enable PAM Access

- RedHat
 - Files
 - /etc/pam.d/system-auth
 - /etc/pam.d/password-auth
 - Add after last account entry
 - account required pam_access.so
- Debian
 - Files
 - /etc/pam.d/login
 - /etc/pam.d/sshd
 - Uncomment
 - account required pam_access.so

This enables /etc/security/access.conf

/etc/security/access.conf

- Format
 - Allow/Deny : user or group : from
- Evaluated in order
 - + :root :ALL allow root from anywhere
 - + :mscott :10.21.32.2 allow mscott from machinee
 - + : (sales) :ALL allow group sales from anywhere
 - :ALL :ALL deny all others

PAM Password Quality Checks

- Is it a dictionary word?
- Is it a palindrome?
- Did only the case change?
- Is it too similar to the old one?
- Is it just a rotated version of the old one?
- Is it too simple?
- Are there more than 5 different characters?
- Does it contain the username or gecos?

/etc/security/pwquality.conf

- minlen **minimum length**
 - difok **difference from old**
 - gecoscheck **no gecos**
 - dictcheck **no words**
 - usercheck **no username**
 - maxrepeat **repeat limit**
 - minclass **min # classes**
- **Classes**
 - Positive is a max credit
 - Negative is a minimum
 - dcredit **digits**
 - ucredit **upper case**
 - lcredit **lower case**
 - ocredit **other**

Password quality examples 1

- Required
 - dcredit = 2
 - lcredit = 0
 - minlen = 10
- Proposed
 - r2wrexac too short (8+1)
 - r2wrexal OK (8+2)
 - r2wre32 too short (7+2)
- Required
 - dcredit = -2
 - lcredit = 0
 - minlen = 10
- Proposed
 - r2wrexacpw only one digit
 - r2wrexal too short
 - r2wrexal32r OK

Password quality examples 2

- Required
 - dcredit = 2
 - ucredit = 1
 - lcredit = 0
 - minlen = 10
- Proposed
 - r2wrexac OK (8+1+1)
 - r2wrexal OK (8+2)
 - r2wRe32 OK (7+2+1)
- Required
 - dcredit = -2
 - lcredit = 0
 - minlen = 10
 - minclass = 3
- Proposed
 - r2wrexacpw only one digit
 - r2wrexal32r only 2 classes
 - R2wrexal32r OK

Configuration Management

Linux System Administration
Fall 2023

How to manage multiple machines

- Scripts that configure users and services
 - Very flexible, can accommodate heterogeneity
 - Ad hoc procedures can be confusing
- Configuration Management tools
 - Declarative desired state of each machine
 - Knows common tasks like user & network configuration
 - Manage large number of machines

CM goals

- idempotent
 - Applying multiple times does not create duplicates
- cross platform
 - Works across flavors of Linux or Unix
 - Some also support Windows
- distributed
 - Apply to all managed machines

Commonly used CM systems

System	Initial Release	Implementation	Configuration	Web site
CFEngine	1993	C	custom	cfengine.com
Puppet	2005	Ruby	custom	puppet.com
Chef	2009	Ruby	Ruby	chef.io
Salt	2011	Python	YAML	saltstack.com
Ansible	2012	Python	YAML	ansible.com

and many more...

https://en.wikipedia.org/wiki/Comparison_of_open-source_configuration_management_software

Using a CM

- Pros
 - Consistent configuration on all hosts
 - Declarative syntax
 - Cross platform
 - Remote management
 - Easier to document
- Cons
 - Errors can take down many hosts
 - CM differ in approach and terminology
 - Manual config/CM conflicts makes ad hoc changes difficult

CM User Example

- Puppet

```
user { "mscott":  
    uid => "2001",  
    ensure => present,  
    comment => "Michael Scott",  
    gid => "mscott",  
    groups => ["managers"],  
    shell => "/bin/bash",  
    home => "/home/mscott",  
    managehome => true,  
}
```

- Ansible

```
name: User Michael Scott  
ansible.builtin.user:  
    name: mscott  
    uid: 2001  
    state: present  
    comment: "Michael Scott"  
    group: mscott  
    groups: managers  
    shell: /bin/bash  
    home: /home/mscott  
    create_home: yes
```

Installing Puppet

- Install package on all machines
 - dnf install puppet
 - apt install puppet
- Install modules
 - puppet module install puppet-network
- Apply a puppet manifest
 - puppet apply dmusers.pp

Puppet Modules

- Public repo <https://forge.puppet.com>
 - Quality and age varies by module
- Builtin modules for basic things
 - files and directories
 - users & groups
- Puppet can be run as client/server
 - Requires setting up puppetserver and certs
 - Manifests often just copied and then applied

Ansible Concepts

- Control node used to manage network
- Managed nodes
 - Accessed via ssh
 - Does not have ansible installed
- Module (code to accomplish tasks)
- Playbook (things to accomplish)
 - YAML file
 - Task (what we want to accomplish)

Installing Ansible

- Ansible need only be installed on the control node

```
dnf -y install ansible or apt -y install ansible
```

- Access to managed hosts is by ssh
 - Set yourself up for password-less logins
- Configure ansible
 - /etc/ansible/ansible.cfg **add** pipelining=true
 - /etc/ansible/hosts **configure** saclass hosts

Applying ansible

- Show hosts in saclass group
 - ansible saclass --list-hosts
- Check connectivity
 - ansible saclass -m ping
- Running privileged commands
 - ansible-playbook -K umask.yaml
 - -K prompt for the BECOME password

Play settings

name: Name of the play

hosts: Hosts to apply this play to

remote_user: The ssh user on the managed node

become: Run command on managed node as another

become_user: User to run remote command as

tasks: What to do

Running a play

```
vlakkies@machinee $ ansible-playbook -K umask.yaml
BECOME password:

PLAY [default umask]
*****
TASK [Gathering Facts]
*****
ok: [100.64.41.6]
ok: [100.64.41.3]
ok: [100.64.41.2]
ok: [100.64.41.4]
ok: [100.64.41.1]

TASK [Set /etc/profile.d/umask.sh] ****
changed: [100.64.41.3]
changed: [100.64.41.6]
changed: [100.64.41.1]
changed: [100.64.41.2]
changed: [100.64.41.4]

PLAY RECAP ****
100.64.41.1      : ok=2    changed=1    unreachable=0    failed=0     skipped=0    rescued=0    ignored=0
100.64.41.2      : ok=2    changed=1    unreachable=0    failed=0     skipped=0    rescued=0    ignored=0
100.64.41.3      : ok=2    changed=1    unreachable=0    failed=0     skipped=0    rescued=0    ignored=0
100.64.41.4      : ok=2    changed=1    unreachable=0    failed=0     skipped=0    rescued=0    ignored=0
100.64.41.6      : ok=2    changed=1    unreachable=0    failed=0     skipped=0    rescued=0    ignored=0
```

Security considerations

- Relies on ssh to access remote machines
 - Requires one credential for access
 - Requires one credential for sudo access
 - An action should require ONE password
- Small site: Use sysadmin credentials
- Large site: Some reasonable compromise

Ansible builtin modules

- `ping` verify basic connectivity
- `user` manage user accounts
- `group` manage user groups
- `file` manage files and properties
- `copy` copy files
- `template` copy a file based on a template
- `systemd` manage systemd services
- `cron` manage crontab and cron.d entries
- `dnf` manage packages using dnf
- `apt` manage packages using apt

Booting and Services

Linux System Administration
Fall 2023

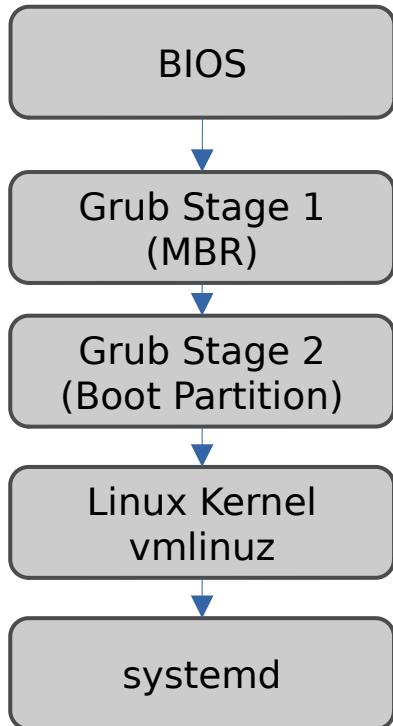
Bootstrapping

- Loading initial software onto the hardware

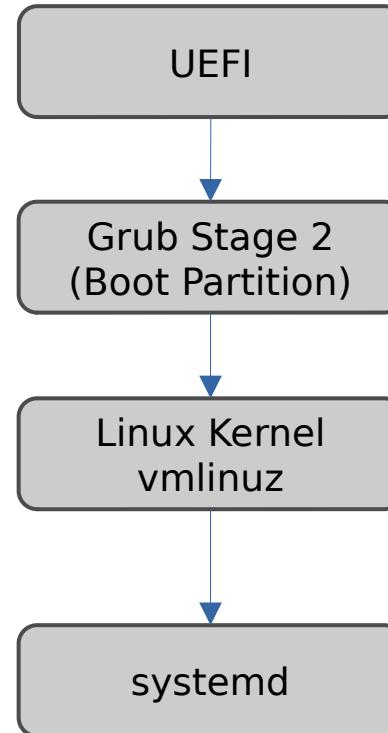


Modern Bootstrap

- Legacy BIOS



- UEFI Boot



Boot Methods

- BIOS
 - **Basic Input/Output System**
- UEFI
 - **Unified Extensible Firmware Interface**
- PXE
 - **Preboot Execution Environment**
 - Part of UEFI

Bootloaders

- loadlin
 - Original linux bootloader (obsolete)
- lilo and elilo
 - **l**inux **l**oader (legacy)
- GRUB
 - **G**Rand **U**nified **B**ootloader
- syslinux
 - group of loaders for CD, USB, etc.

GRUB

- BIOS started from ROM
 - Stage 1 loaded from Master Boot Record
 - Stage 2 loaded from /boot
- UEFI started from ROM
 - Stage 2 loaded from /boot
- Reads /boot/grub/grub.cfg (made from /etc/grub.d)
 - Which kernel to load
 - Parameters to pass to the kernel
 - Runs vmlinuz

Loading the kernel

- vmlinuz compressed Linux kernel
- initramfs or initrd.img
 - Initial RAM disk file system for kernel
- config kernel configuration stuff
- System.map kernel crash symbol maps

Kernel parameters

- Check running kernel `cat /proc/cmdline`
- Edit by `e` on GRUB boot prompt
 - `root=` root file system (`UUID=` or `/dev/????`)
 - `ro` mount read only
 - `quiet` quiet boot
 - `systemd.unit=` set systemd target
 - `rd.break` start shell instead of systemd
 - `init=` process to launch

Managing processes

- init
 - Traditional UNIX mother of all processes
 - PID=1
 - Used by BSD and legacy and rebel Linux distros
 - Runlevels 0-6
 - rc scripts
- systemd
 - New parallel process manager
 - PID=1
 - RedHat, Debian, ...
 - targets
 - units
 - too many other things

init runlevels

- Runlevel use
 - 0 system halt
 - 1 single user
 - 2 multi-user
 - 3 multi-user with networking
 - 4 no standard
 - 5 graphical login
 - 6 reboot
- Set as init N
- Legacy Linux/System V
 - Scripts in /etc/init.d
 - start, stop, restart, reload
 - Linked to /etc/rcN.d
 - /etc/rc2.d/K80apache2
 - Kills apache2 at runlevel 2
 - /etc/rc3.d/S20apache2
 - Start apache2 at runlevel 3
 - Sort determines order
- BSD somewhat different

systemd concepts

- target
 - More flexible than runlevels
 - Names have meaning
 - rescue
 - multi-user
 - graphical
 - Dependency tree
 - Can be user defined
- unit
 - Defines a service, device,...
 - Explicit dependencies
 - Wants start in parallel
 - Requires start after
 - Also Before and After
 - WantedBy when to start

systemd files

- `/lib/systemd` **systemd code**
- `/lib/systemd/system` **unit and target files**
- `/etc/systemd` **configuration files**
- `/etc/systemd/system/* .wants` **enabled units**
 - **symbolic links to** `/lib/systemd/system`
- `/bin/systemctl` **control program**

Example unit: apache2.service

```
[Unit]
Description=The Apache HTTP Server
After=network.target remote-fs.target nss-lookup.target
Documentation=https://httpd.apache.org/docs/2.4/

[Service]
Type=forking
Environment=APACHE_STARTED_BY_SYSTEMD=true
ExecStart=/usr/sbin/apachectl start
ExecStop=/usr/sbin/apachectl graceful-stop
ExecReload=/usr/sbin/apachectl graceful
KillMode=mixed
PrivateTmp=true
Restart=on-abort

[Install]
WantedBy=multi-user.target
```

Example Unit: ssh.service

```
[Unit]
Description=OpenBSD Secure Shell server
Documentation=man:sshd(8) man:sshd_config(5)
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=/etc/default/ssh
ExecStartPre=/usr/sbin/sshd -t
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/usr/sbin/sshd -t
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify
RuntimeDirectory=sshd
RuntimeDirectoryMode=0755

[Install]
WantedBy=multi-user.target
Alias=sshd.service
```

Managing systemd

- `systemctl` **action unit**
 - `start` **start immediately**
 - `stop` **stop immediately**
 - `restart` **stop&start immediately**
 - `enable` **set to start on boot**
 - `disable` **do not start on boot**
 - `status` **show status**
 - `is-enabled` **will it start on boot?**
 - `is-active` **is it running?**
 - `daemon-reload` **restart systemd**
 - `isolate` **activate target**
 - `show units`
 - `list-dependencies` **show tree**
 - `mask` **prohibit unit from starting**
 - `default = isolate` **default**
 - `set-default` **set default target**
 - `get-default` **show default target**

Troubleshooting systemd

- Check if a service is running and enabled
- Stop, start, restart, enable, disable service
- Recognize, get and set the default target
- Isolate a target

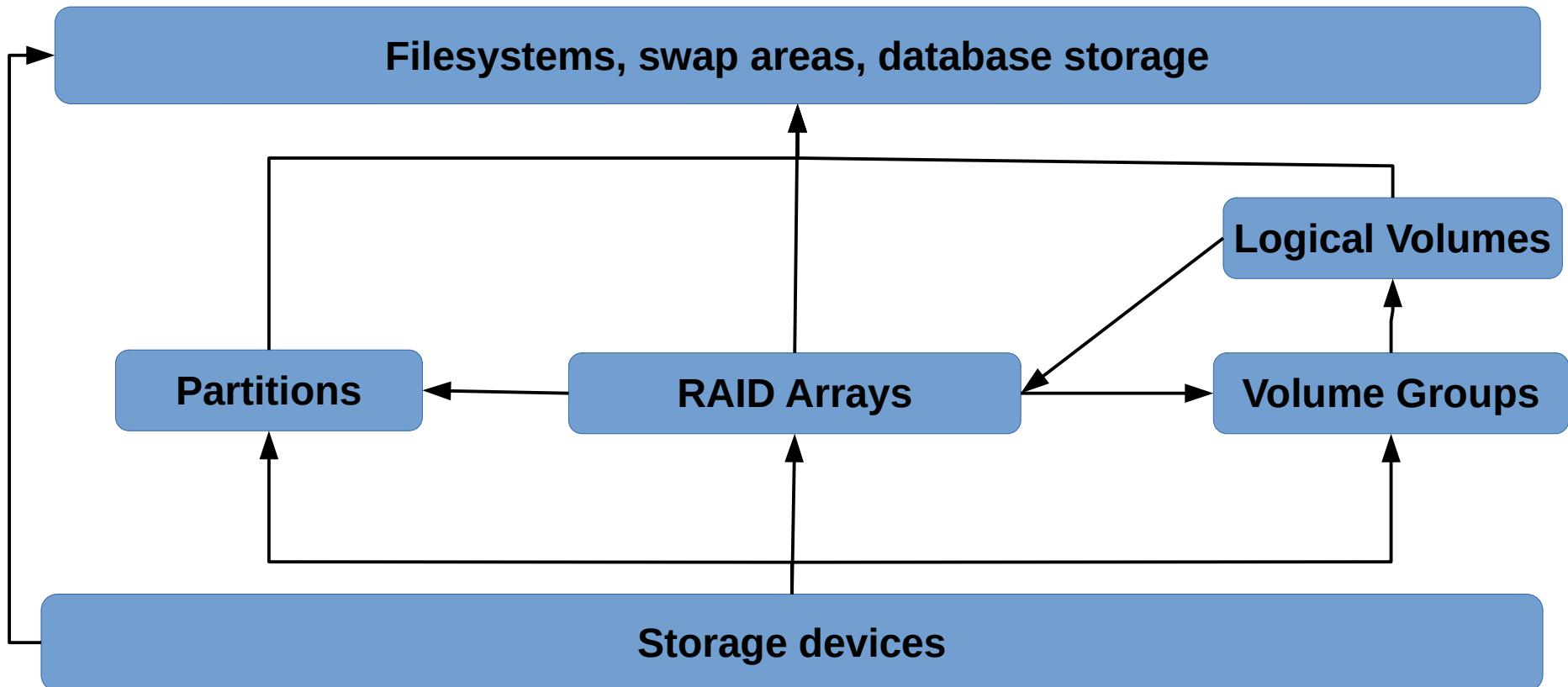
Filesystems

Linux System Administration
Fall 2023

Filesystems

- On Unix/Linux, everything is a file
- Files are organized in a directory tree
- A file system is what stores files
 - File systems integrate into the directory tree
- A disk is a physical storage device
 - Storage blocks are called sectors
 - Disk can be divided into partitions
 - Disks can be grouped in RAID arrays

Storage Management

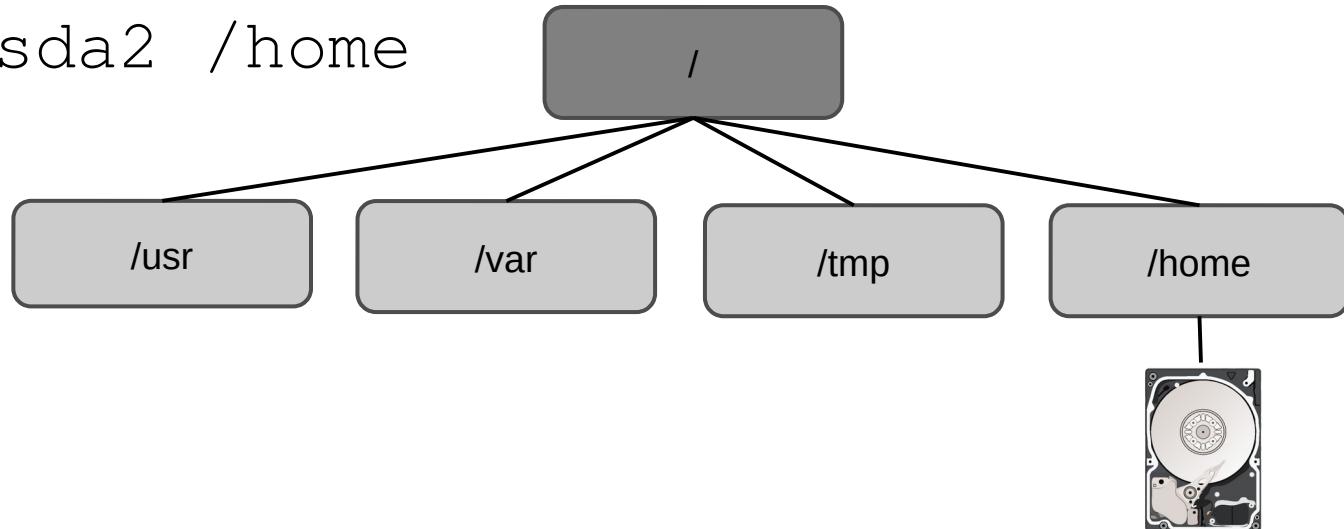


Useful filesystem separation

- /boot separate partition for grub
 - /boot/efi separate partition for grub/efi
- /tmp or /scratch temporary storage
- /home user directories
- /var contains cache, mail, run, etc
- /var/log contains system logs
- /var/lib/mysql database storage

Mounting filesystems

- mount device dir
 - The mount point must be an existing directory
 - The file system becomes part of the tree
 - Existing sub-tree becomes inaccessible
 - mount /dev/sda2 /home



Unix file system concepts

- Derived from the original Fast File System (FFS)
 - Disks are block devices
 - Directory tree structure
 - Directory entries point to inodes
 - inodes contain file characteristics
 - Type, permissions, times, etc.
 - Pointers to data blocks & indirection
 - Superblocks, cylinder groups, fragments
- Journaling added much later

File system features to consider

- Stability & robustness
 - Detecting bit rot
 - Safe handling of power or system failure
- Ability to store very large and very small files
- Performance
 - Read and write speed
 - Recovery time after failure
- Snapshots, replication, compression, encryption, etc.

Popular file systems

- UFS Unix File system (FFS)
- ext2/ext3/ext4 Extended File System
 - ext3 added journaling (backwards compatible)
- XFS extent based file system
- ZFS incorporates RAID and volume manager
- Btrfs B-tree file system("ZFS lite")
- tmpfs volatile memory file system
- *FAT* File Allocation Table (1977 floppy disks - current USB)

File systems can be mixed

- vfat for /boot/efi to simplify booting
- ext4 for / good for general purpose use
- xfs for /home to store large media files
- BrtFS for /var/lib/mysql for snapshots
- NFS4 for /var/mail network shared storage
- File system type and mount point are seamless
 - df -T or cat /etc/fstab shows the detail

Device Naming

- `/dev/hdX` Legacy hard drives
- `/dev/sdX` SCSI (SATA, SAS, USB) drives
- `/dev/srX` CD/DVD drives
- X is in the order of discovery a, b, c, d, ..
- Partitions are appended numerically
 - `/dev/sdc1` First partition (1) on third drive (c)

Persistent file names

- **UUID/GUID Universally/Globally Unique ID**
 - Stored on disk in file system superblock
 - `/dev/disk/by-uuid/*` symbolic link to `/dev/*`
- `blkid` shows UUID and other information
- UUID can be referenced in `/etc/fstab`

/etc/fstab

- Disks to mount on boot (or `mount -a`)
 - **device** **mount** **type** **options** **backup** **check**
 - `/dev/sda1 / ext4 errors=remount-ro 0 1`
 - **device** name of the device
 - `/dev/sda1 UUID=4c3f0e41-9bec-449c-8c57-95b5a4cb9927 /dev/mapper/root`
 - **mount point** where this go in the directory tree
 - `/` or `/home` or `/var/lib/mysql`
 - **type** file system type
 - `ext4` or `xfs` or `tmpfs` or `nfs4` or `swap`
 - **options** mount options
 - `errors=` or `ro` or `rw` or `sw` and may others
 - **backup** (obsolete - used by `dump` - set to zero)
 - **check** file system check order
 - `0=no fsck, 1=fsck first (file root), 2=fsck next`

Filesystem commands

- `mount` mount a filesystem in the tree
- `umount` unmount a filesystem
- `fsck` file system check
 - Generic front end for `fsck.ext3`, `fsck.xfs`, etc
- `mkfs` create a new file system (**D.F.I.U.**)
 - Generic front end for `mkfs.ext3`, `mkfs.xfs`, etc
- `badblocks` check drive for bad blocks (low level)

Scripting should be done with caution

- If something goes wrong, the remainder of the script can do a LOT of damage
- If you forget to update some parameters, you can destroy the system



MY RESPONSE WHENEVER ANYONE ASKS
ME TO MESS AROUND WITH FILESYSTEMS

Filesystem specific commands

- ext2/ext3/ext4
 - tune2fs
 - set options
 - manage quotas
 - resize2fs
 - shrink or expand
 - dumpe2fs
 - show filesystem data
- xfs
 - xfs_admin
 - set options
 - xfs_growfs
 - expand
 - xfs_info
 - show filesystem data
 - xfs_quota
 - manage quotas

Disk Partitions

- Subdivides a disk (or storage system)
 - Presents just like a disk to the file system
 - Simplifies backups
 - Provides a degree of protection
 - Difficult to change later
- Stored on the disk
 - MBR Master Boot Record
 - GPT Globally Unique ID Partition Table

Managing partitions

- Partition programs
 - fdisk (text UI)
 - text UI
 - parted
 - text UI
 - better for fixing stuff
 - gparted
 - GUI version
- DFIU
 - Make sure you are in the right disk!!!!!!!!!!
 - Moving partitions can destroy file systems
 - Expanding OK, shrinking typically not

Partition Table

- Partition properties
 - Primary/Secondary
 - Bootable flag
 - Start and end sectors
 - Size (sectors)
 - Type
- Partition types (hex)
 - 82 Linux swap
 - 83 Linux
 - 8e Linux LVM
 - a5 FreeBSD
 - fd Linux RAID
 - 86 NTFS

Machine A Partition Table

- **# fdisk -l /dev/sda**

```
Disk /dev/sda: 16 GiB, 17179869184 bytes, 33554432 sectors
Disk model: Virtual disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: A7854EC3-3878-49A5-AFFF-3E7976B0F137
```

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	2099199	2097152	1G	EFI System
/dev/sda2	2099200	10487807	8388608	4G	Linux filesystem
/dev/sda3	10487808	29378559	18890752	9G	Linux LVM

- **# df -kh**

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	1022M	7.0M	1016M	1%	/boot/efi
/dev/sda2	4.0G	315M	3.7G	8%	/boot
/dev/mapper/r1-root	8.0G	1.4G	6.7G	17%	/
devtmpfs	4.0M	0	4.0M	0%	/dev
tmpfs	1.8G	0	1.8G	0%	/dev/shm
tmpfs	731M	27M	705M	4%	/run
tmpfs	366M	0	366M	0%	/run/user

RAID

- Redundant Array of Inexpensive Disks
 - Hardware chips or entire subsystems
 - Software RAID kernel module (`md`)
- RAID concepts
 - **Stripe** spread data across devices to improve throughput
 - **Mirror** save redundant information to survive drive failures

Types of RAID

- JBOD **J**ust a **B**unch **O**f **D**isks
- 0 Stripes data across 2 disks
- 1 Mirrors data to 2 disks
- 5 Stripe and mirror, survive 1 lost disk (min 3)
- 6 Stripe and mirror, survive 2 lost disks (min 4)
- 0+1 Mirror of stripes (4 disks)
- 1+0 Stripe of mirrors (4 disks)

RAID considerations

- RAID does not reduce the need for backups
- Hot spares are nice, but reduces capacity
- RAID5 many writes required hurt performance
- RAID5 vulnerable to desync on power fail
 - Battery backup improves robustness+performance
- Software RAID is inexpensive but slow

Linux Software RAID

- **md (multiple disks)**
- Devices are physical disks or partitions
- Managed by mdadm
- **Creating an array**

```
mdadm --create /dev/md/name --raid-devices=3  
          --level=5 /dev/sdf1 /dev/sdg1 /dev/sdh1
```

- **Making the array persistent**

```
mdadm --verbose --detail --scan>/etc/mdadm.conf
```

Logical Volume Manager (LVM)

- Physical Volume
 - Disk drives, partitions or RAID subsystems
- Volume Group
 - Collection of physical volumes
- Logical Volume
 - Logical device (partition within volume group)

LVM Pros and Cons

- Software RAID
 - More flexible, but slower than hardware
- Logical volumes can be readily resized
 - Readily add more physical volumes
 - More disks or partitions
 - Most filesystems can expand, some can shrink
- Snapshots

LVM Commands

- Physical Volume
 - pvcreate
 - pvdisplay
 - pvchange
 - pvck
 - pvremove
 - pvresize
- Volume Group
 - vgcreate
 - vgdisplay
 - vgchange
 - vgck
 - vgremove
 - vgextend
 - vgscan
- Logical Volume
 - lvcreate
 - lvdisplay
 - lvchange
 - lvscan
 - lvremove
 - lvextend

Simple LVM Recipe

- Label device(s)
 - `pvcreate /dev/sdb1`
- Create volume group
 - `vgcreate SA /dev/sdb1`
- Create logical volumes
 - `lvcreate -n tmp -L 1G SA`
 - `lvcreate -n mysql -L 4G SA`
- Make file systems
 - `mkfs -t xfs /dev/mapper/SA-tmp`
 - `mkfs -t xfs /dev/mapper/SA-mysql`
- Mount filesystems
 - `mount /dev/mapper/SA-tmp /tmp`
 - `mount /dev/mapper/SA-mysql /var/lib/mysql`

Swap

- Disk blocks that can be used to swap memory blocks to in virtual memory systems
 - Can be useful for large occasional memory hogs
 - Heavily using swap kills performance
- Linux swap commands
 - `mkswap` set up device for swap
 - `swapon` show or enable swap
 - `swapoff` disable swap

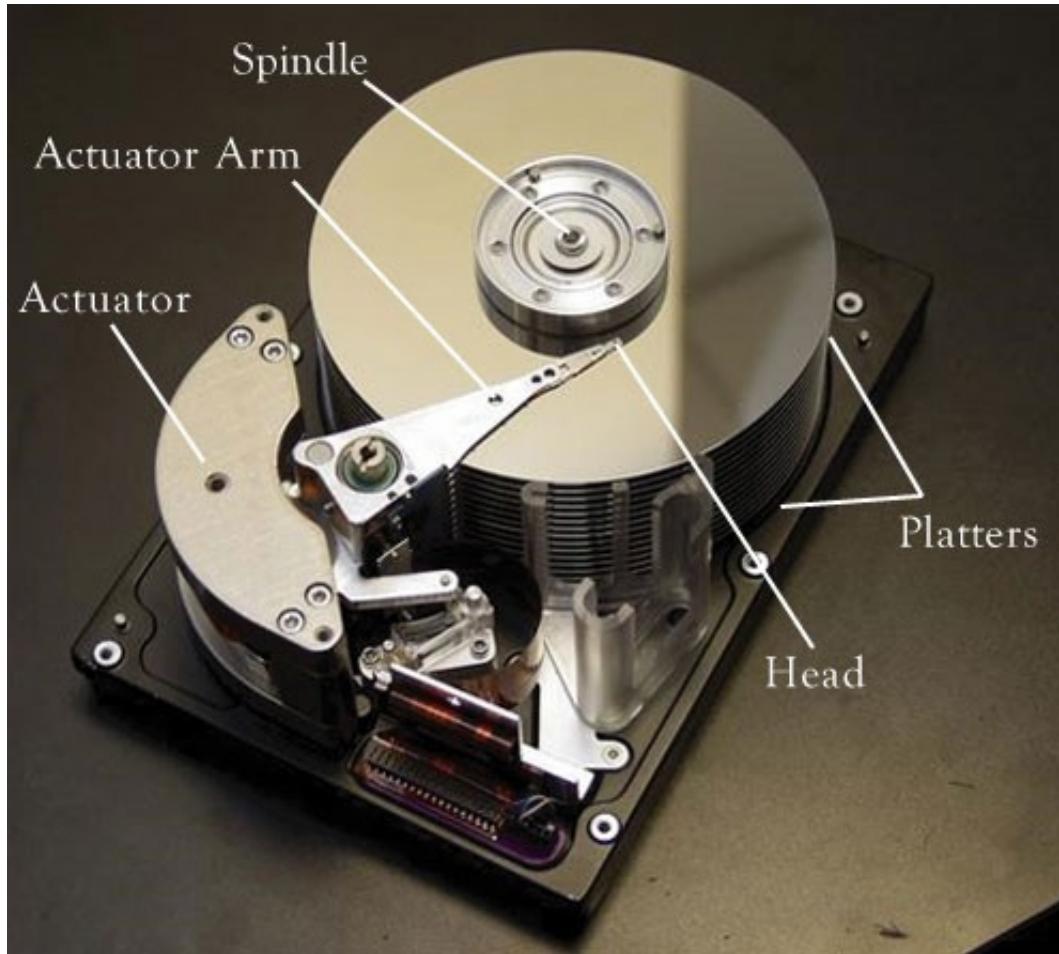
Network File System

- NFS is the native Unix Network File System
 - NFSv4 is the current versions
- Allows a server to export a subtree to clients
- Clients mount the subtree like a drive
 - `mount emc:/vol1 /home`
- Automount can mount or unmount on demand

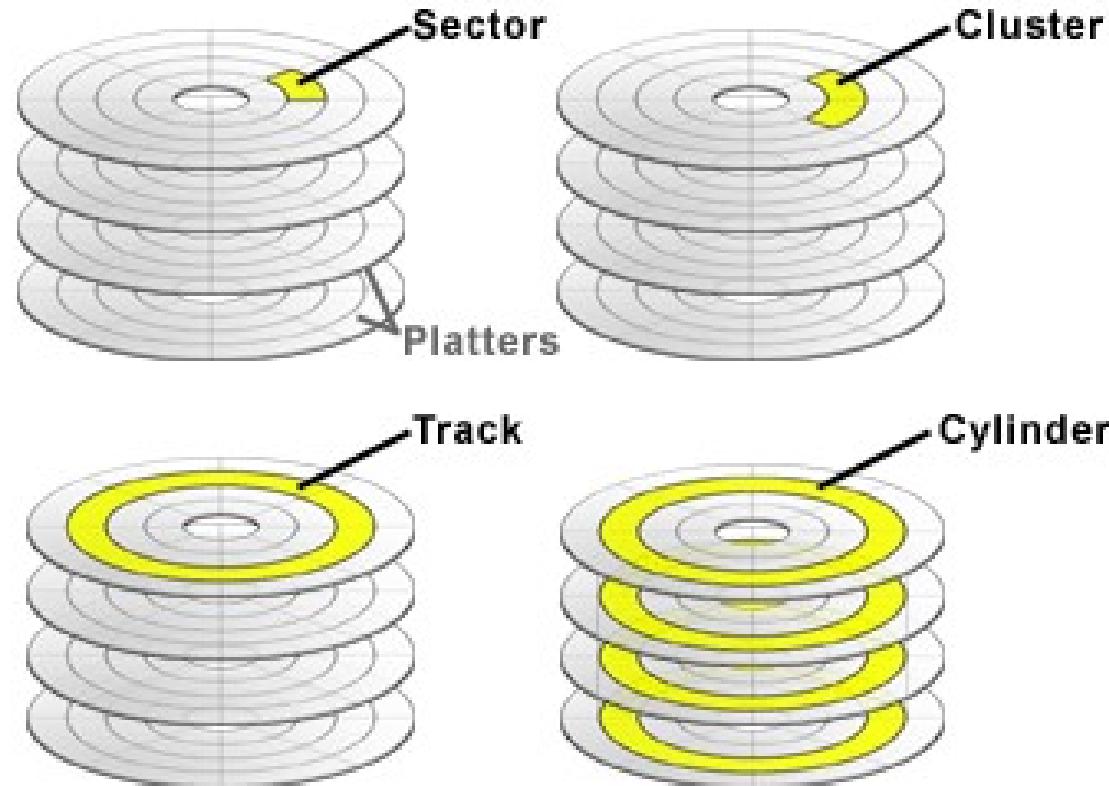
Disk drive types

- Mechanical
 - Slower 4-15 ms
 - Larger capacity
 - Less expensive
 - Sequential access must faster than random
 - Unlimited rewrites
 - Failure often predictable
- Solid State
 - Faster 0.1 ms
 - Lesser capacity
 - More expensive
 - Fast access regardless of pattern
 - Finite rewrites
 - Often fails without warning

Mechanical Hard Drives

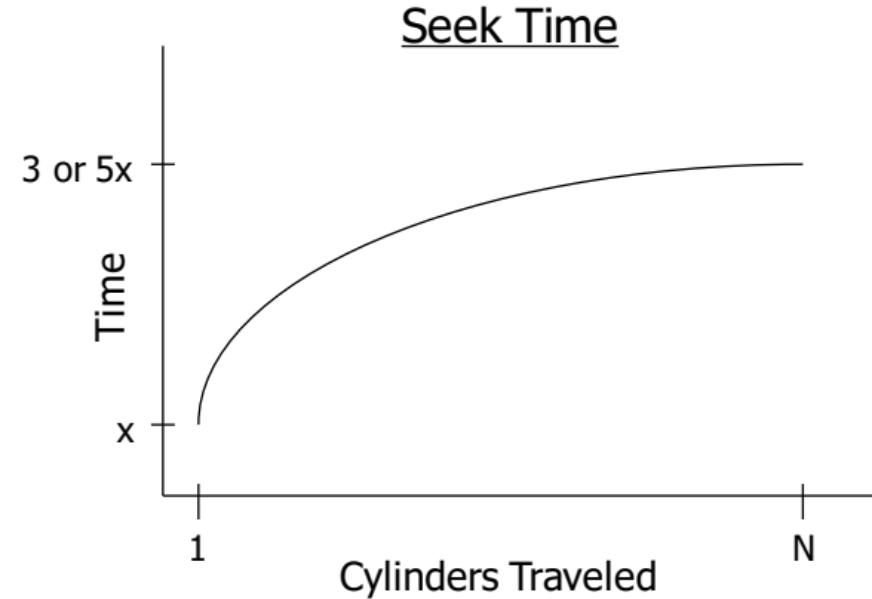


Mechanical Hard Drive Organization



Mechanical Hard Drive Properties

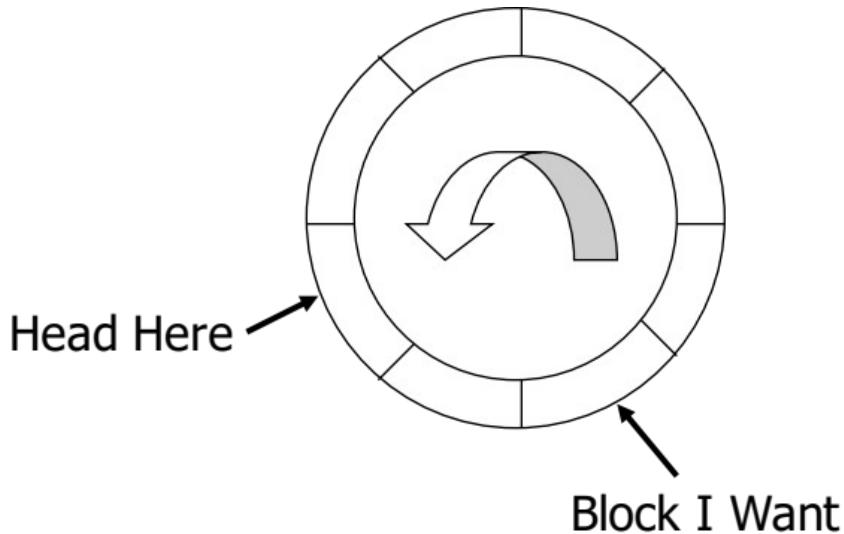
- Access time factors
 - Seek time
 - Acceleration
 - Transit
 - Deceleration
 - Settling
 - Rotational delay
 - Transfer time



Disk scheduling algorithms

- First come, first served
- Shortest Seek Time First
- SCAN (elevator algorithm)
- C-SCAN (one way elevator)
- C-LOOK (stops at first/last requested cylinder)

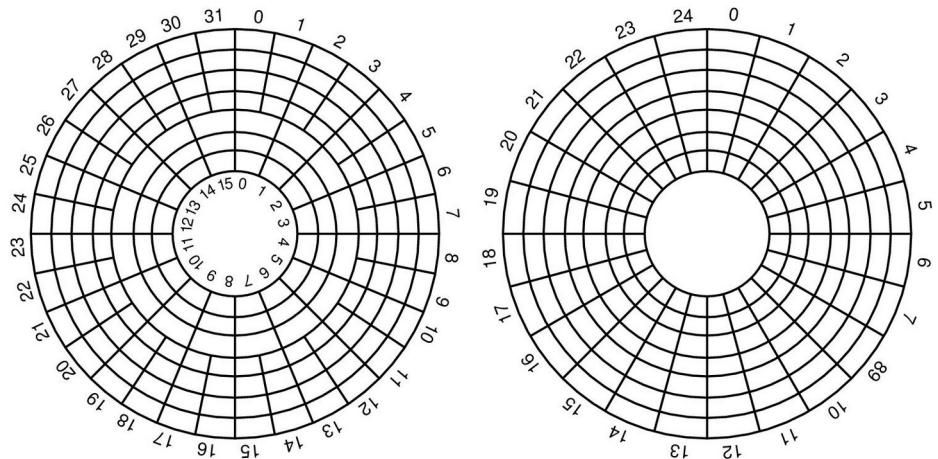
Rotational Delay



HDD Spindle [rpm]	Average rotational latency [ms]
4,200	7.14
5,400	5.56
7,200	4.17
10,000	3.00
15,000	2.00

Transfer Rate

- Transfer rate
 - Speed of media
 - Linear density
 - Faster on the outside
 - Potentially more sectors per track
 - Track total/rotation



Disk Interface Alphabet Soup

- Legacy Interfaces
 - IDE (ATA, PATA) *Integrated Drive Electronics*
 - SCSI *Small Computer System Interface*
- Current Interfaces
 - SATA *Serial ATA [1.5-6.0 Gb/s]*
 - SAS *Serial Attached SCSI [3-22.5 Gb/s]*
 - PCIe (NVMe) *Peripheral Component Interconnect Express*
 - USB *Universal Serial Bus*

Overall performance

- Access time = seek + rotation + transfer
- Write performance may be slower for SSD
 - Erase before write
- Caching
- Read-ahead
- RAID

Disk failures

Why you should have good backups

- MTBF *Mean Time Between Failure*
- S.M.A.R.T *Self-Monitoring, Analysis and Reporting Technology*
- Bad blocks, bad cells and self-healing
 - Low level formatting
- Drive types
 - Value - big and slow
 - Mass-market - fast and hot
 - NAS - robust and cool
 - Enterprise - fast and expensive

Logging

Linux System Administration
Fall 2023

Importance of Logging

- History of events
 - Kernel events
 - System reboots
 - System errors
 - Service start/stop
 - Startup warnings
 - cron jobs
- Audit trail
 - logins
 - refused logins
 - sudo commands
 - files accessed
 - invalid or refused
 - mail messages
 - DHCP requests

Logging types

- **syslog**
 - IETF standard
 - includes remote logging
 - facility.severity
- **systemd-journal**
 - logging by systemd
 - binary format
- Custom logging
 - Apache
 - access
 - error
 - sendmail
 - sudo

syslog

- rsyslog newer version
 - IETF standard
 - Systematic logging
time host [PID] message
- Configured by /etc/rsyslog.conf and /etc/rsyslog.d/*.conf
 - selectors facility.level action

syslog selectors

- facilities (*=all except mark)
 - auth authorization
 - authpriv sensitive auth
 - cron cron daemon
 - daemon system daemons
 - kern kernel
 - localX local messages
 - X = 0-7
 - mark regular time
 - mail mail
 - syslog internal
 - user default
- level (minimum) *=all
 - emerg system panic
 - alert action required
 - crit critical error
 - err error condition
 - warning could be a problem
 - notice pay attention
 - info informational
 - debug testing
 - none nothing

syslog actions

- `filename` Append to filename (- means no sync)
- `@hostname` Send to remote host
- `@ipaddress` Forward to IP port UDP 514
- `@@ipaddress` Forward to IP port TCP 514
- `| fifo` Write to named pipe
- `user1, user, ...` Write message to user consoles
- `*` Write to all user consoles
- `^prog;template` Send to program
- `~` Discard

syslog example (Machine C)

```
auth,authpriv.*          /var/log/auth.log
*.*;auth,authpriv.none  -/var/log/syslog
cron.*                  /var/log/cron.log
daemon.*                -/var/log/daemon.log
kern.*                  -/var/log/kern.log
user.*                  -/var/log/user.log

# Split mail
mail.*                  -/var/log/mail.log
mail.info               -/var/log/mail.info
mail.warn               -/var/log/mail.warn
mail.err                /var/log/mail.err

# Catchall
*=info;*=notice;*=warn;auth,authpriv.none; \
cron,daemon.none;mail.none  -/var/log/messages
```

systemd-journal

- systemd attempt to take over logging
- Binary journal for quick retrieval
 - More difficult to manage than text files
- Access with `journalctl`
 - See output in `systemctl status`
- Works in tandem with syslog
 - Log messages also forwarded to syslog

journalctl examples

- Show all boots
 - `journalctl --list-boots`
- Show logs as they arrive (follow)
 - `journalctl -f`
- Show entries for a service (`sshd`)
 - `journalctl -u sshd` (**RedHat**)
 - `journalctl -u ssh` (**Debian**)

Other logging

- Apache
 - Configured in <virtualHost>
 - TransferLog - standard log format
 - CustomLog - custom fields and order
 - Stored in /var/log/httpd or /var/log/apache2
- sudo
 - Configured in /etc/sudo_logsrvd.conf
 - Stored in /var/log/secure or /var/log/auth.log

logger

- Program to add syslog entries (from scripts)
- `logger -p facility.level message`
 - `logger -p daemon.info "Start foo"`
 - `logger -n log.dm.com "Machine C boot"`
- Also useful for testing syslog configs
- systemd-journal has to be different
 - `echo 'Start foo' | systemd-cat -t myapp -p info`

Managing logs

- Logs are stored in `/var/log` or `/var/log/*`
- Log aging set by site policy
- `logrotate`
 - `/etc/logrotate.conf` and `/etc/logrotate.d/*`
 - Starts new log file daily, weekly, monthly, etc.
 - Keeps older logs as `.1`, `.2.gz`, `.3.gz`, ...
- By default `systemd` journal grow without limit
 - Limit in `/etc/systemd/journald.conf` with `SystemMaxUse`

Central Logging

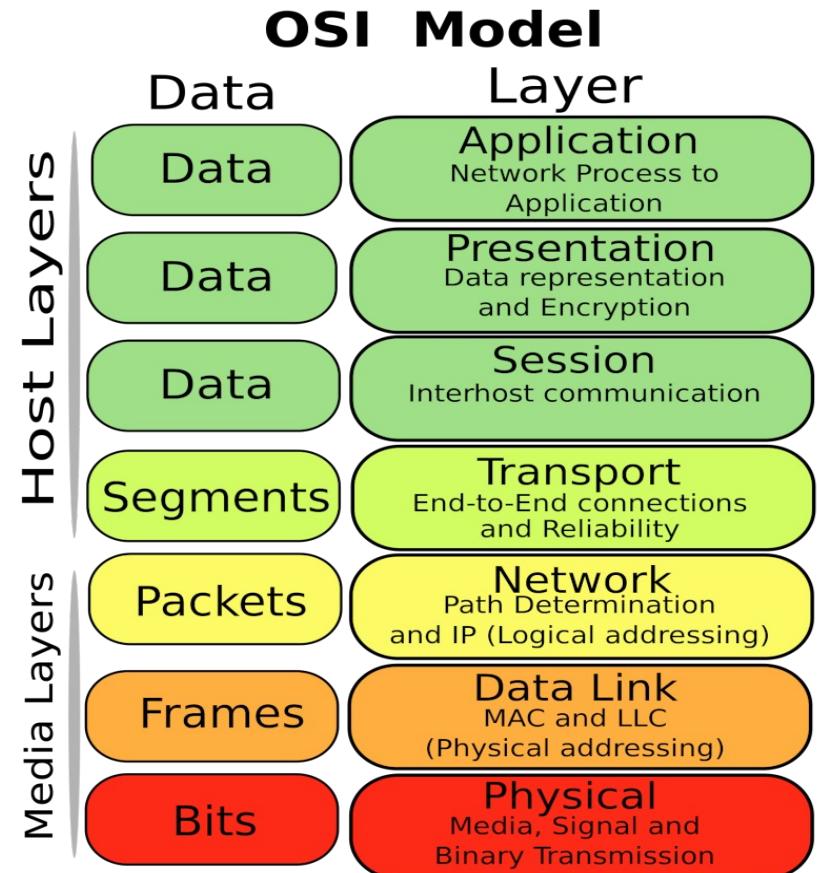
- It doesn't take much to drown in logs
 - Roll your own with syslog and scripts
 - ELK : Elasticsearch, Logstash, Kibana
 - Network Management Software (e.g. Observium)
- Make sure all your systems agree on time
 - NTP to synchronize clocks
 - Consistent time zone

Networking

Linux System Administration
Fall 2023

Practical Networking

- Most network traffic is IPv4 or IPv6
 - TCP or UDP
- Most traffic is carried by Ethernet, Fiber or Microwave
 - 802.*



Evi's Extended OSI Model

- Networking requires many policy decisions
 - Security vs. convenience
 - Interfacing with other networks and groups
 - Common protocols



Common IP Tools

- ping
 - Tests end-to-end connectivity
- traceroute or (or tracert on Windoze)
 - Shows the route the packet takes
- ip addr or ifconfig
 - IP configuration
- ip route or route or netstat -nr
 - Configure or show routing

Ethernet

- Developed in 1970s
 - Commercialized 1980s
 - Shared medium ("ether")
 - Retransmit on collision
- Layer 2
 - 48bit MAC address
 - Data frames
- Clear winner over competitors
 - Marketing over performance
- Initially used taps on coaxial cable with terminators
 - 10base2 & 10base5
- Largely replaced by copper unshielded twisted pair
 - 10baseT, 100baseT, 1000baseT (RJ45)
 - >1G typically fiber
 - Current max 400Gb/s

Generic Network Hardware

- hub, repeater, concentrator
 - Retransmits all frames on all ports
 - No configuration
- switch, bridge
 - Retransmits frames based on MAC address (layer 2)
 - Self-configuration (Spanning Tree Protocol, ...)
- router
 - Routes packets based on IP header (layer 3)
 - Requires configuration

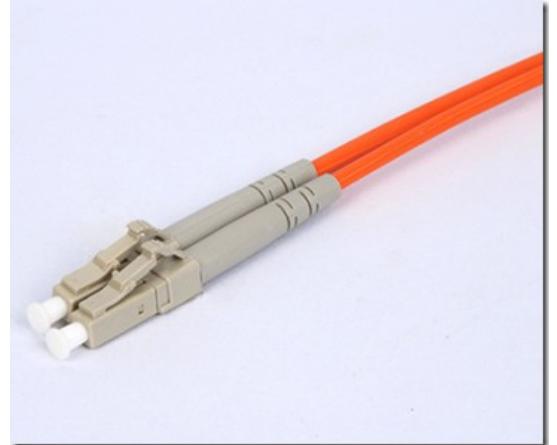
Copper (Unshielded Twisted Pair)

- RJ-45
- Cat5, Cat6, Cat7, ...
- T568B colors
- Straight/Cross-over
- Max length ~100m
- Easy field termination
- Power-over-ethernet
- Autonegotiation
- Failure modes
 - Lightning & RF
 - Rodents



Fiber Optics

- LC, SC, ST, ...
- Single/multi mode
 - OS1,OM1,OM2,OM3
 - Color denotes type
- Limited bending radius
- Many km runs
- Difficult field termination
- Failure modes
 - Backhoe
 - Bend radius



Radio Frequency (UHF & microwave)

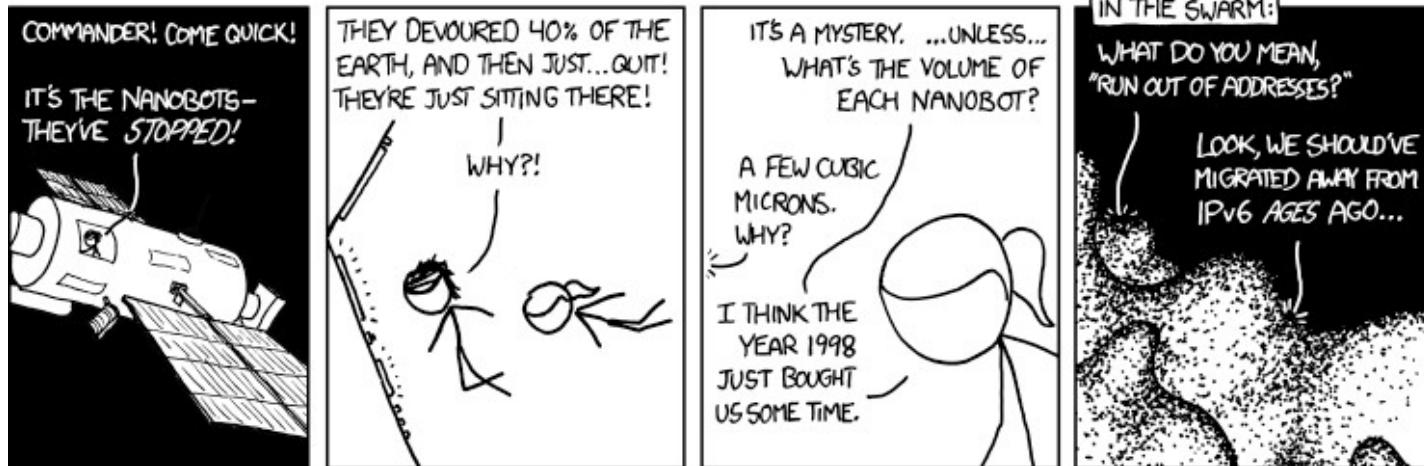
- 300MHz-1GHz
 - << 1Mb/s speeds
 - Beyond line of sight
- 2.4 GHz & 5GHz WiFi
 - Unlicensed
- 3-30 GHz licensed
 - Usually point-to-point
- 60 GHz point-to-point
- Generally 802.11*
 - Speed of light matters in very long shots
 - SSID, modes, modulation...
- Antennas
 - Polarization
 - Omni, Sector, High gain
- Noise and fade

Internet Protocol (IP)

- Packet switched network
 - Introduced in 1981
 - Replaced older circuit switched networks
- Advantages
 - Decentralized, hierarchical configuration
 - Robust, dynamic routing
 - IPv4 supports 4,294,967,296 unique addresses

IP Versions

- IPv1, IPv2, IPv3 - Early development versions (messing with TCP/IP)
- IPv4 - What most networks are based on (32 bit, 4 billion addresses)
- IPv5 - Experimental Quality of Service addition
- IPv6 - Next generation IP (128 bit addresses)



IPv4 Addresses

- Every device must have a unique address
- IPv4 use 32 bits organized as 4 octets
 - 01100100 01000000 00000000 00101010 = 100.64.0.42
 - subnet part host part
 - All 32 bits are used to identify the device
 - The subnet part determines what can be reached directly
 - Typically written as 100.64.0.42/24

What is a subnet?

- Part of a greater (interconnected) network
- Group of IP addresses that can directly communicate
 - Can find device at Layer 2 via Ethernet or WiFi
 - Extend using hub, switch, bridge
 - Same leading bits (subnet address) but unique hosts bits
- Subnets are connected by routers
 - Determines forwarding based in IP address

Classfull Inter-Domain Routing

- Prior to 1993 routers assumed classes
 - Class A = 1.x.x.x to 126.x.x.x
 - 126 nets of 16,777,326 hosts each
 - Class B = 128.0.x.x – 191.255.x.x
 - 16,384 nets of 65,536 hosts each
 - Class C = 192.0.0.x – 223.255.255.x
 - 2,752,512 nets of 256 hosts each
 - Class D = 224.x.x.x-239.x.x.x
 - Multicast
 - Special cases 0.x.x.x, 127.x.x.x, 100.64.x.x

Classless Inter-Domain Routing (CIDR)

- Breaks network/host at any bit position
- Example: 10.30.20.0/24
 - Class A address (10.x.x.x)
 - /24 makes it a class C (256 addresses)
- Example: 10.30.20.96/29
 - No classed equivalent
 - 8 addresses
 - Netmask `11111111 11111111 11111111 11111000` = 255.255.255.248
- Extended the life of IPv4
 - Many more unique networks

Netmask

- Mask used to isolate subnet and host parts
 - All 1 bits must be left of all 0 bits

/16	11111111	11111111	00000000	00000000	= 255.255.0.0
/24	11111111	11111111	11111111	00000000	= 255.255.255.0
/29	11111111	11111111	11111111	11111000	= 255.255.255.248
- Network address = Host Address & Netmask
 - Sets host bits to all zeroes
- Hosts on same subnet can communicate directly
 - All others need to be routed

Special host part addresses

- Network address (host part all zeroes)
 - 100.64.42.0/24 01100100 01000000 00101010 00000000
- Broadcast address (host part all ones)
 - 100.64.42.255/24 01100100 01000000 00101010 11111111
- Hosts cannot be assigned network or broadcast address
 - except /31 and /32
- Other *special* addresses are just by convention
 - Gateway typically the lowest address or highest valid address
 - 100.64.0.254/24 or 100.64.0.1/24
- `ipcalc` is convenient for doing network calculations

Address Resolution Protocol (ARP)

- Layer 2 - Layer 3 address translation
 - IPv4 address to MAC address
 - Requires no user configuration
- Contains Layer 2 & 3 addresses
 - OPER field determines action
 - request: What is the MAC address for this IP?
Sent to broadcast MAC address
 - reply: This is the MAC address for this IP.
Sent to request MAC address
 - Exchange requires just one packet each way
 - Other hosts may learn IPv4:MAC from the exchange

Common IP Protocols

- **ICMP Internet Control Message Protocol**
 - Diagnostic - echo, unreachable, TTL exceeded
 - Control - redirect
- **UDP User Datagram Protocol**
 - Unreliable, unordered packets (datagrams)
 - Connection-less between IP:port pairs
- **TCP Transmission Control Protocol**
 - Reliable, ordered, error-checked continuous stream
 - Connection oriented between IP:port pairs

Ports (added by TCP & UDP)

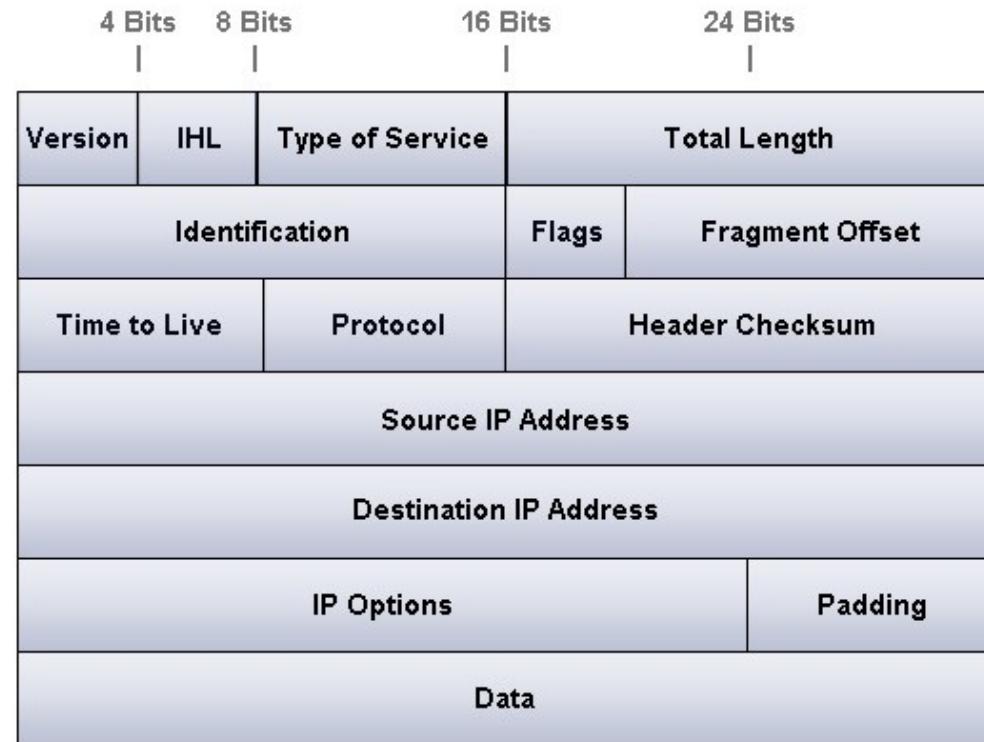
- 16 bit number 1-65535
- Defines a process to talk to
- Ports 1-1023 are *well-known* ports
 - Defined in /etc/services
 - Used by programs to give ports symbolic names
- Ephemeral ports (>1023) mostly used for user programs
- Notation IP:port
 - 100.64.0.42:22 means 100.64.0.42 port 22
 - TCP or UDP

Example /etc/services

ftp-data	20/tcp	snmp	161/udp
ftp	21/tcp	snmp	161/tcp
ssh	22/tcp	snmp-trap	161/tcp
telnet	23/tcp	snmp-trap	161/ucp
smtp	25/tcp	https	443/tcp
domain	53/tcp	syslog	514/udp
domain	53/udp	imaps	993/tcp
http	80/tcp	pop3s	995/tcp
pop3	110/tcp	nfs	2049/tcp
ntp	123/udp	nfs	2049/udp
imap2	143/tcp	mysql	3306/tcp

Anatomy of an IP packet

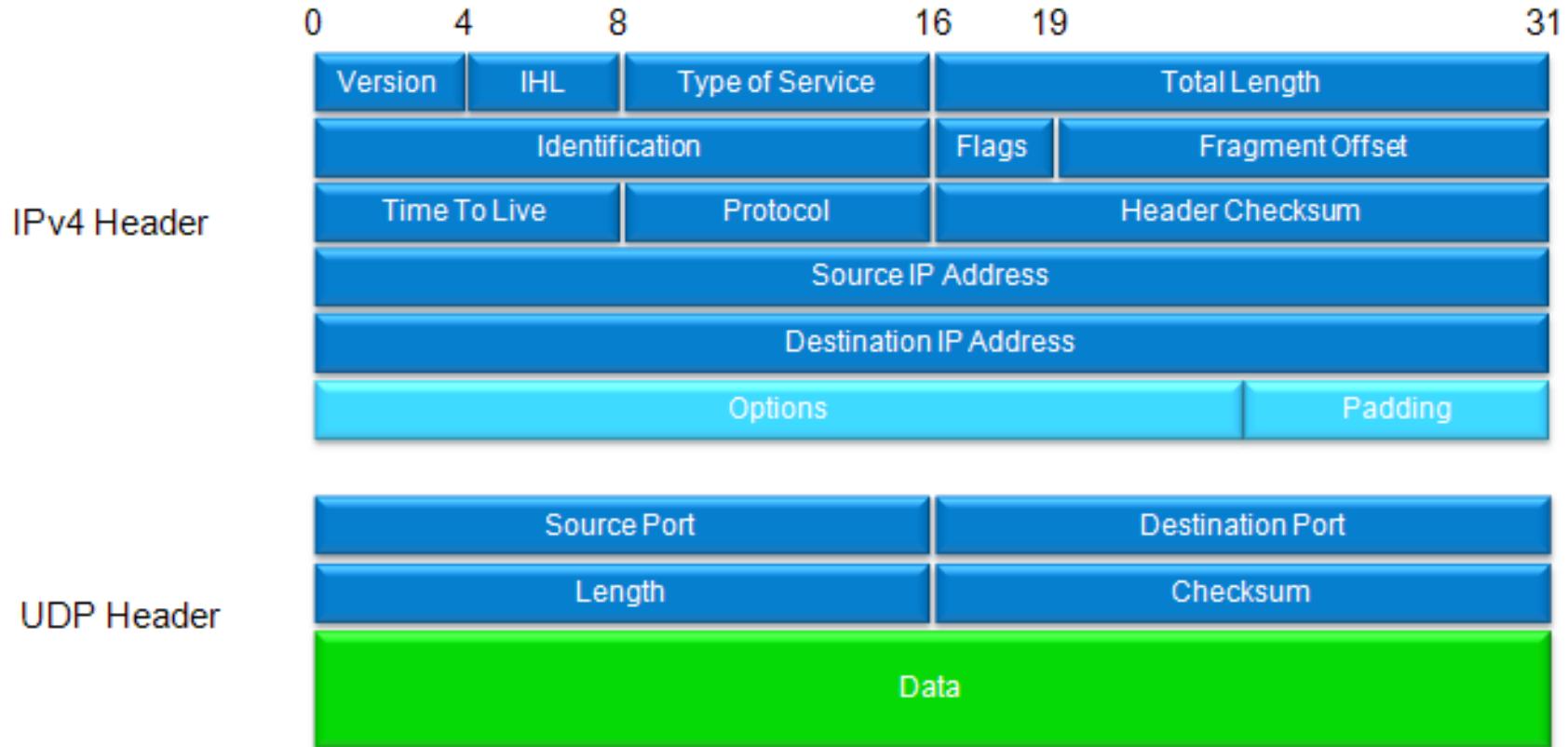
- Always big-endian
- Version=4
- IHL=IP Hdr Len
- Type of Service
 - Min delay
 - Max throughput
- Flags & Frag Off
 - Large packets
- Protocols add additional header in data section



Anatomy of an ICMP Packet

IP Datagram						
	Bits 0–7	Bits 8–15	Bits 16–23	Bits 24–31		
IP Header (20 bytes)	Version/IHL	Type of service	Length			
	Identification		<i>flags and offset</i>			
	Time To Live (TTL)	Protocol	Checksum			
	Source IP address					
	Destination IP address					
	Type of message	Code	Checksum			
ICMP Header (8 bytes)	Header Data					
ICMP Payload (optional)	Payload Data					

Anatomy of a UDP Packet



UDP Applications

- Applications that transmit discrete chunks
 - DNS - Domain Name Service
 - NTP - Network Time Protocol
- Applications with time sensitive information
 - Voice over IP (VoIP)
- Applications that add robustness
 - NFSv2, mosh

Transmission Control Protocol (TCP)

- Bidirectional virtual circuit
 - Input = Output
 - Traffic arrives in order
 - Retransmit, congestion, etc happens invisibly
 - No inherent framing
- Used by many higher level protocols
 - Terminal connections (ssh, telnet)
 - Data transfer (HTTP, FTP, scp, rsync)

TCP stuff you should know

- TCP exchanges
 - 3-way handshake: SYN, SYN-ACK, ACK
 - Termination: FIN, FIN-ACK
 - Reset: RST, RST-ACK
- Flow control and sequence numbers
- Keep-alive
- TCP based protocols need to define segments
 - Example: HTTP uses \r\n

Configuring an IPv4 interface

- Must have
 - IP address
 - Netmask
 - Default gateway
- Nice to have
 - DNS server address
 - NTP server address
- Example:
 - IP 100.64.0.42/24
 - GW 100.64.0.254
 - DNS1 128.138.240.1
 - DNS2 128.138.130.30
 - NTP 132.163.97.1
 - time-a.www.nist.gov

Manual Interface Configuration

- ip link
 - Show available interfaces
- ip addr add 100.64.42.6/24 dev ens192
 - Sets IP address for ens192 to 100.64.42.6
 - Adds route to 100.64.42.0/24 via ens192
- ip route add default via 100.64.42.1
 - Adds default route via gateway 100.64.42.1
 - Reachable on 100.64.42.0/24 using ens192
- ip neigh shows IP and MAC addresses

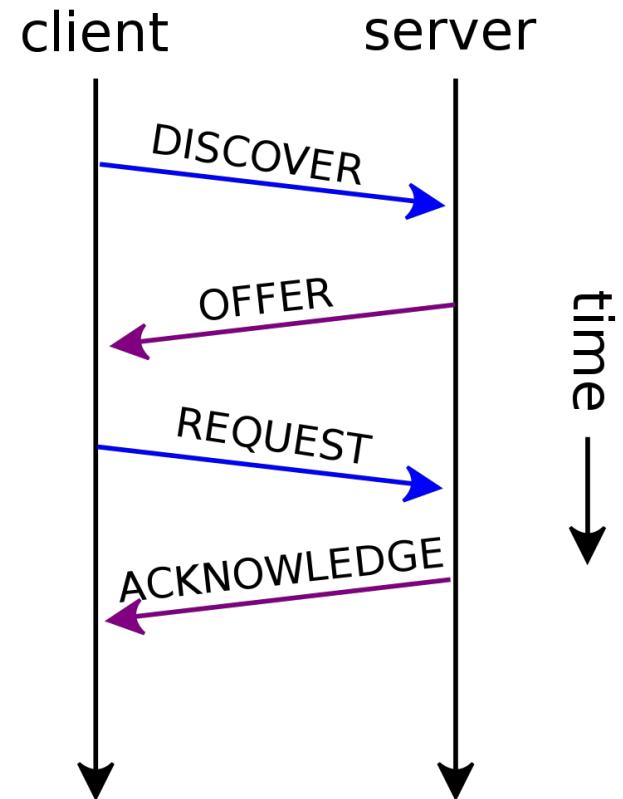
DHCP

(Dynamic Host Configuration Protocol)

- Server assigns unique IP address
 - Centralizes configuration of IP address space
 - Straddles Layer 2 and Layer 3
 - IP from pool or reservation by MAC address
 - IP, netmask and default gateway
- Typically also sets DNS, NTP, etc.

DHCP Exchange

- Broadcast
 - > DISCOVER
 - I need an IP address
- Unicast
 - < OFFER
 - You may use this IP
 - > REQUEST
 - I want to take you up on that
 - < ACKNOWLEDGE
 - You got it



Linux interface names

- Traditional names eth0, eth1, eth2, ...
 - Changes depending order of discovery
- Predictable names
 - eno* (onboard) e.g. eno1
 - ens* (hotplug) e.g. ens192
 - enp* (bus) e.g. enp2s0
 - enx* (MAC) e.g. enx525400d5e0fb
- Names set by udev

Persistent Interface Configuration

- Redhat
 - Set in /etc/sysconfig/network-scripts
 - Configuration by interface, e.g. ifcfg-ens192
- Debian
 - Set in /etc/network/interfaces
 - All interfaces configured in one file

RedHat Interface Example

- */etc/sysconfig/network-scripts/ifcfg-ens192*

```
DEVICE=ens192
ONBOOT=true
BOOTPROTO=static
IPADDR=100.64.42.6
PREFIX=24
GATEWAY=100.64.42.1
```

Debian Interface Example

- `/etc/network/interfaces`

```
auto lo ens192
```

```
iface lo inet loopback
```

```
allow-hotplug ens192
```

```
iface ens192 inet static
```

```
    address 100.64.42.6
```

```
    netmask 255.255.255.0
```

```
    gateway 100.64.42.1
```

NetworkManager

- Developed by RedHat to handle laptops and in particular wireless networking
 - GUI, nmtui and nmcli
- Getting harder to remove
 - Honors /etc/sysconfig/network-scripts
 - RHEL 9 prioritize key file format configurations in /etc/NetworkManager/system-connections/

RedHat Key File Example

- /etc/NetworkManager/system-connections/ens192.nmconnection

```
[connection]
id=ens192
type=ethernet
interface-name=ens192
```

```
[ipv4]
method=manual
address1=100.64.42.6/24,100.64.42.1
```

Routing

- Router connects multiple subnets
- The next hop must be reachable at Layer 2
- Each subnet has an associated target
 - via (IP address of gateway)
 - Packet sent to the MAC address of the gateway
 - interface
 - Packet sent to the MAC address of destination
- Selects **best** match out of all possibilities

DNS

Linux System Administration
Fall 2023

Domain Name System

- Distributed database created to replace text file containing every host name in the world
- Hierarchical naming system
 - top level domains (e.g. .com, .edu, .org, .uk)
 - domains (google.com, colorado.edu, janet.ac.uk)
 - hosts (www.google.com, magellan.colorado.edu)
- Distributed database
 - Authoritative servers for each level
- FQDN - fully qualified domain name

Translating Names to IP addresses

- **/etc/nsswitch.conf sets name resolution**
 - hosts: files dns **selects first /etc/hosts then DNS**
- **/etc/hosts for ad hoc settings**
 - 100.64.42.3 machinec machinec.dundermifflin.com
- **/etc/resolv.conf for DNS servers**
 - search dundermifflin.com
 - nameserver 100.64.42.2
 - nameserver 100.64.42.6

BIND

- Berkeley Internet Name Domain
 - Created early 1980s, released with BSD 4.3
 - BIND 9 released 2000 (actively maintained by ISC)
- named (name daemon)
 - Main configuration in `named.conf`
 - Supports primary and secondary servers
 - DNS includes a transfer mechanism to synchronize servers
 - Views allows one server to do split DNS

dig @ns type fqdn

- Sections
 - Question
 - Answer
 - Authority
 - Additional
- Use -x for reverse DNS

```
dig simlab.colorado.edu
; <>> DiG 9.16.1-Ubuntu <>> simlab.colorado.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8697
;; flags: qr rd ra; QUERY:1, ANSWER:1, AUTHORITY:3, ADDITIONAL:3

;; QUESTION SECTION:
;simlab.colorado.edu. IN A

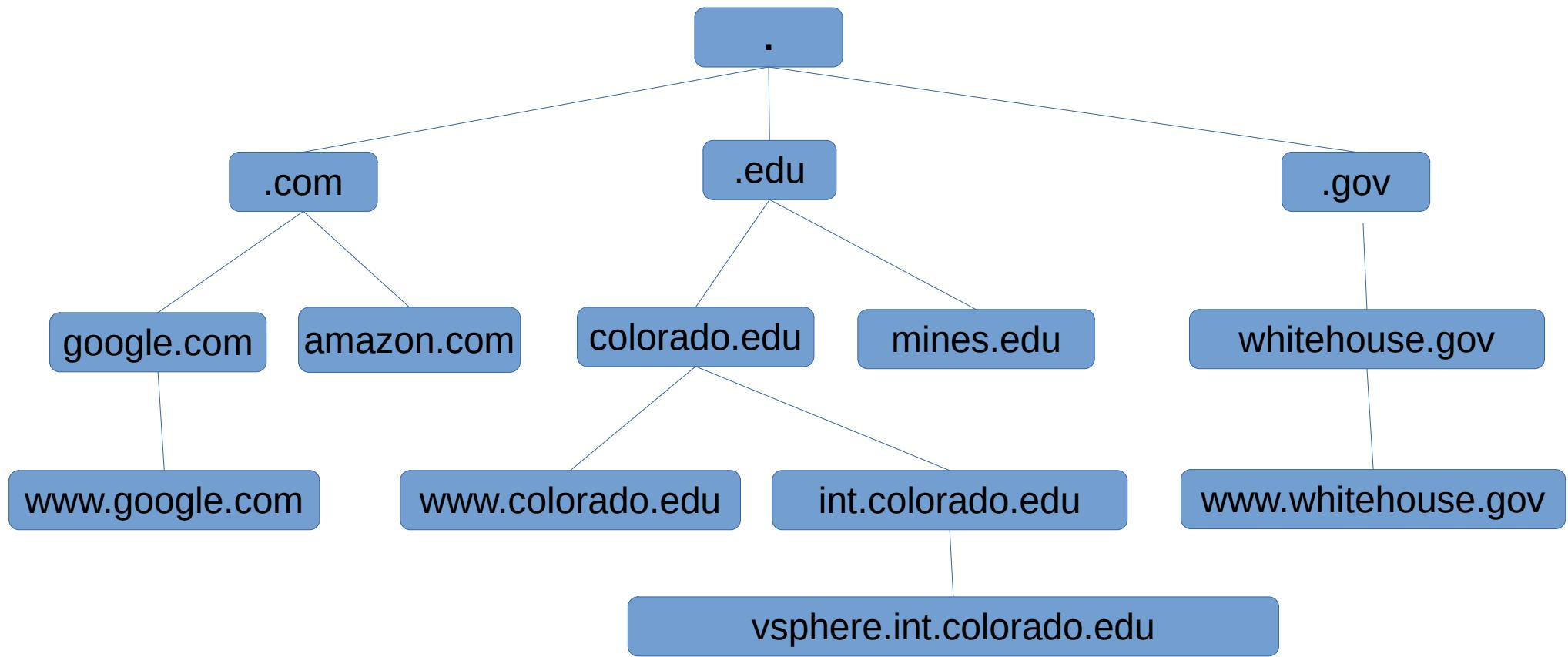
;; ANSWER SECTION:
simlab.colorado.edu. 3600 IN A 128.138.73.46

;; AUTHORITY SECTION:
colorado.edu.3600 IN NS oldduke.colorado.edu.
colorado.edu.3600 IN NS otis.colorado.edu.
colorado.edu.3600 IN NS boulder.colorado.edu.

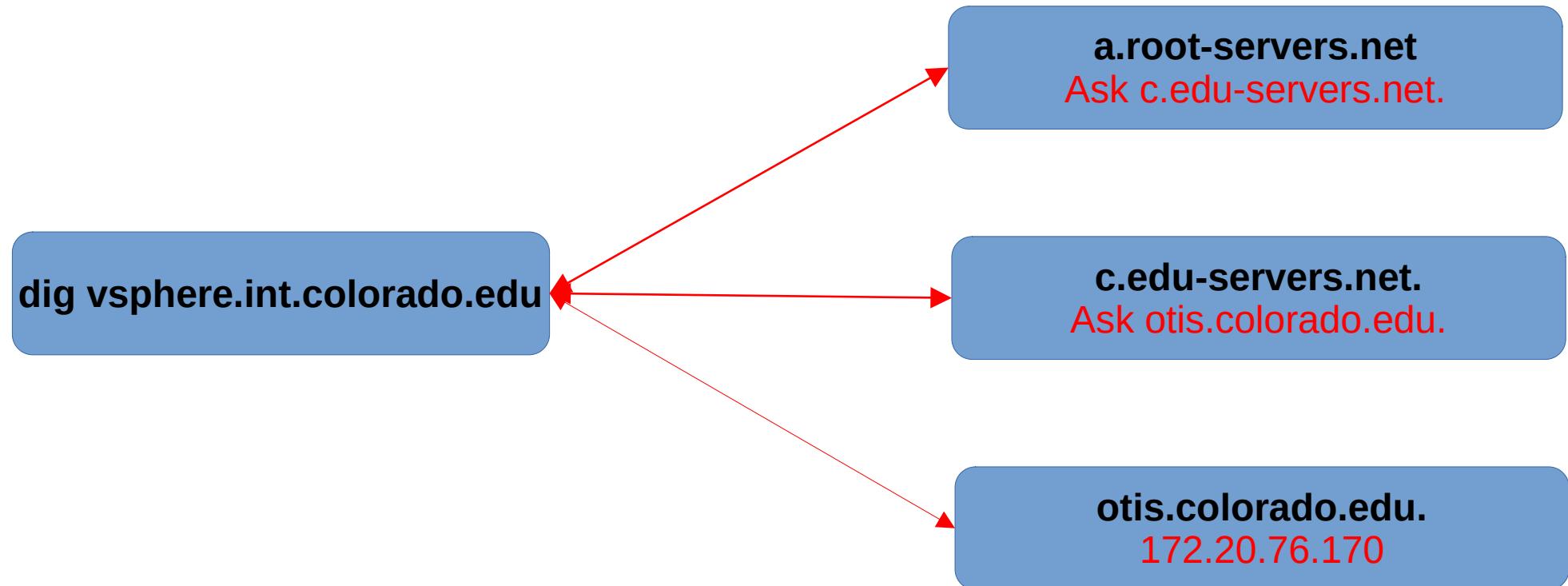
;; ADDITIONAL SECTION:
otis.colorado.edu. 139621 IN A 128.138.129.76
boulder.colorado.edu. 3324 IN A 128.138.240.1
oldduke.colorado.edu. 139621 IN A 128.138.130.30

;; Query time: 23 msec
;; SERVER: 192.168.11.2#53(192.168.11.2)
```

Forward DNS Hierarchy



How DNS works



How DNS works detail

```
dig @a.root-servers.net.  
      vsphere.int.colorado.edu
```

```
;; AUTHORITY SECTION:  
edu. IN NS a.edu-servers.net.  
edu. IN NS b.edu-servers.net.  
edu. IN NS c.edu-servers.net.  
edu. IN NS d.edu-servers.net.  
edu. IN NS e.edu-servers.net.  
edu. IN NS f.edu-servers.net.  
edu. IN NS g.edu-servers.net.  
edu. IN NS h.edu-servers.net.  
edu. IN NS i.edu-servers.net.  
edu. IN NS j.edu-servers.net.  
edu. IN NS k.edu-servers.net.  
edu. IN NS l.edu-servers.net.  
edu. IN NS m.edu-servers.net.
```

```
;; ADDITIONAL SECTION:  
a.edu-servers.net. IN A 192.5.6.30  
b.edu-servers.net. IN A 192.33.14.30  
c.edu-servers.net. IN A 192.26.92.30  
d.edu-servers.net. IN A 192.31.80.30  
e.edu-servers.net. IN A 192.12.94.30  
f.edu-servers.net. IN A 192.35.51.30  
g.edu-servers.net. IN A 192.42.93.30  
h.edu-servers.net. IN A 192.54.112.30  
i.edu-servers.net. IN A 192.43.172.30  
j.edu-servers.net. IN A 192.48.79.30  
k.edu-servers.net. IN A 192.52.178.30  
l.edu-servers.net. IN A 192.41.162.30  
m.edu-servers.net. IN A 192.55.83.30
```

```
dig @c.edu-servers.net. vsphere.int.colorado.edu
```

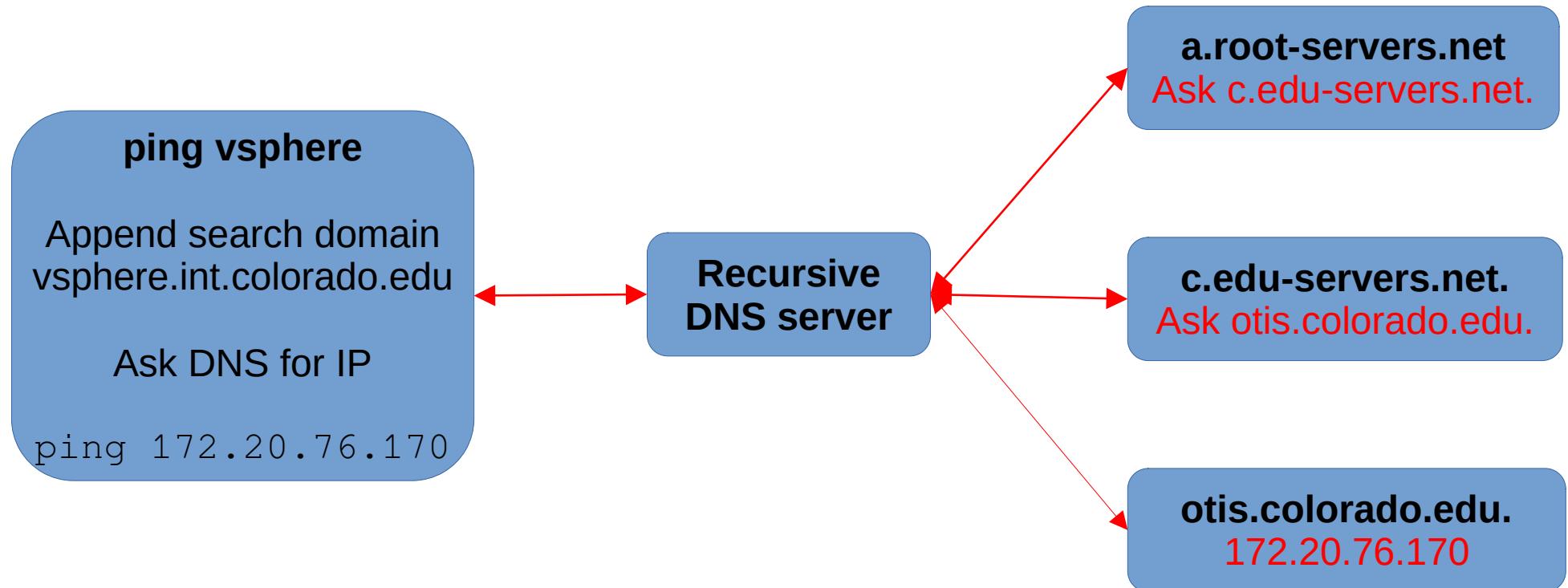
```
;; AUTHORITY SECTION:  
colorado.edu. IN NS boulder.colorado.edu.  
colorado.edu. IN NS otis.colorado.edu.  
colorado.edu. IN NS oldduke.colorado.edu.
```

```
;; ADDITIONAL SECTION:  
boulder.colorado.edu. IN A 128.138.240.1  
otis.colorado.edu. IN A 128.138.129.76  
oldduke.colorado.edu. IN A 128.138.130.30
```

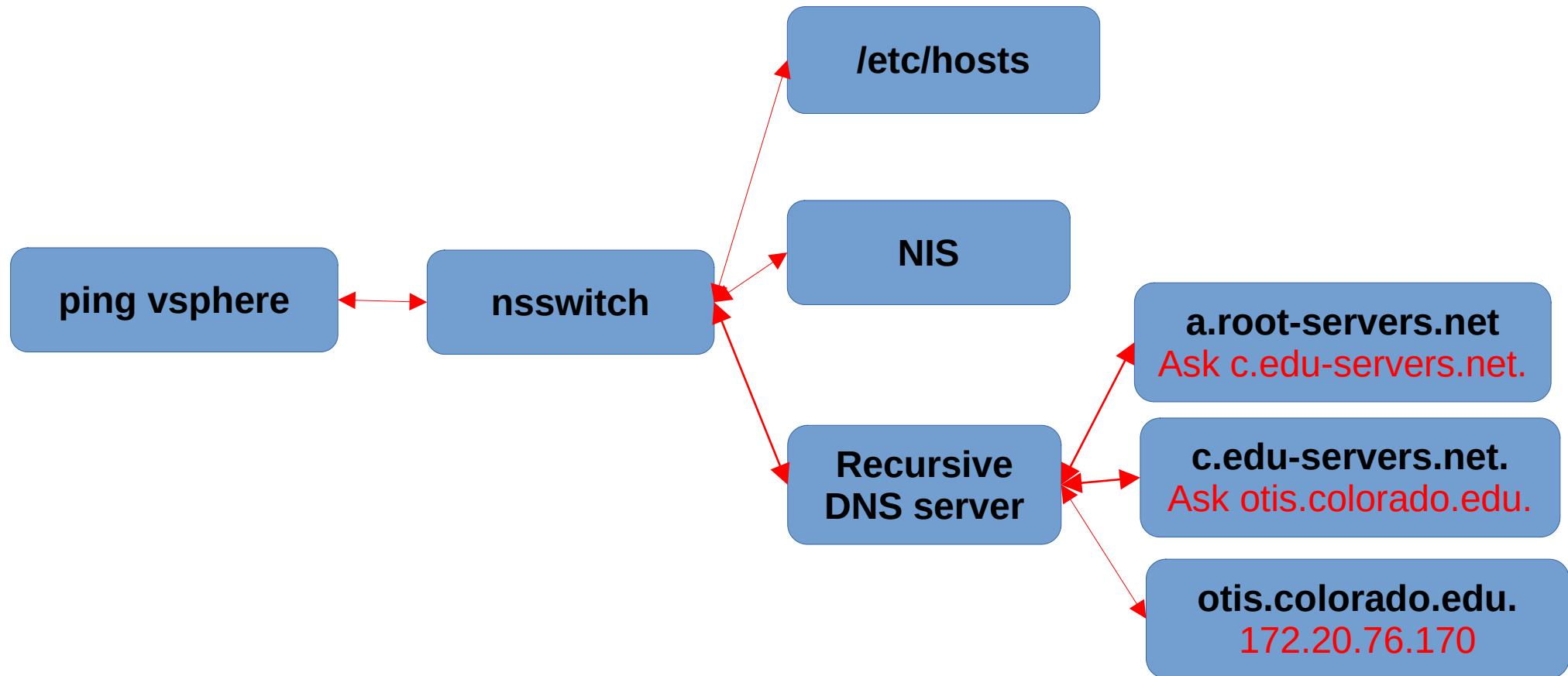
```
dig @otis.colorado.edu. vsphere.int.colorado.edu
```

```
;; ANSWER SECTION:  
vsphere.int.colorado.edu. IN A 172.20.76 .170
```

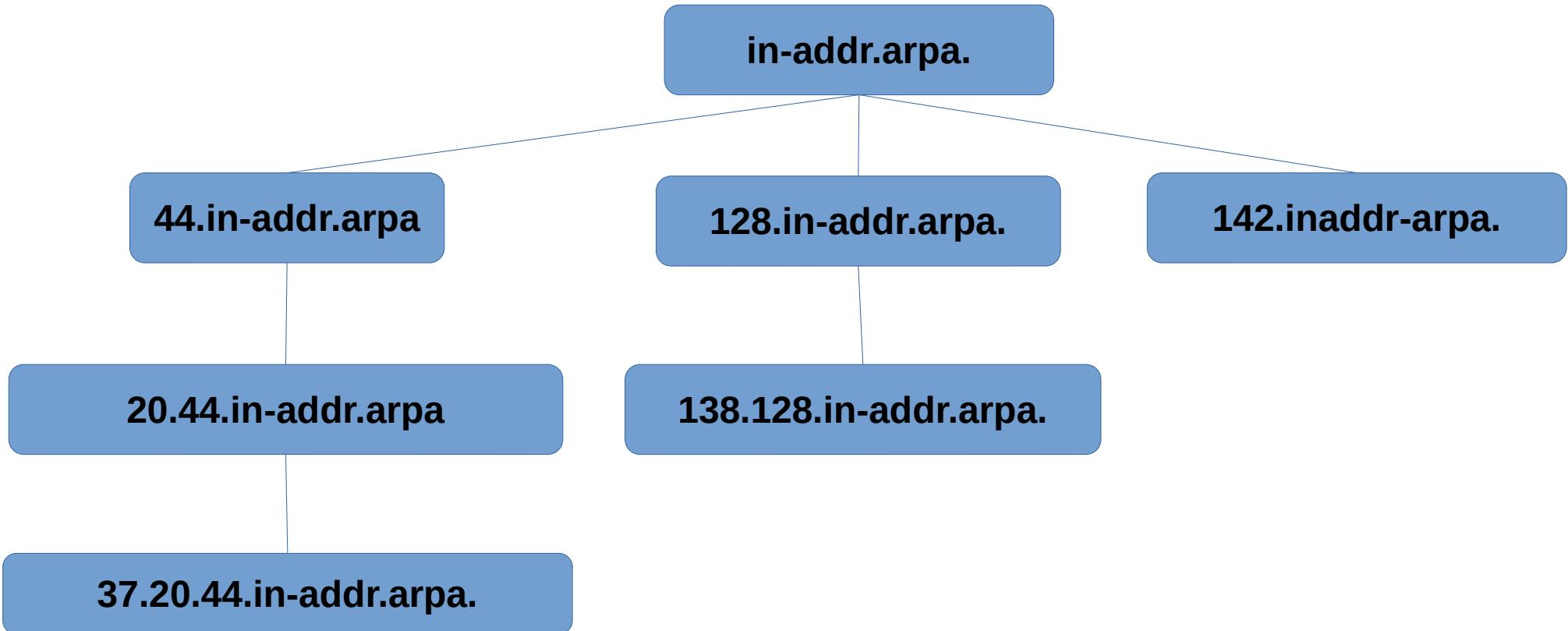
Recursive DNS



Linux/Unix DNS



Reverse DNS



DNS Server Classifications

- Authoritative
 - Has the definitive answer
- Recursive
 - Will work to get you the answer
- Master or Primary
 - Original source of resource records
- Secondary
 - Redundant backups

DNS zone

- A zone is a specific subtree of the DNS namespace
- A zone file in named.conf specifies how to resolve addresses within that zone
 - Resource records for the zone
 - References to other servers
 - Root hints
 - Forwarders

Resource Records

- **SOA** - Start of Authority
- **NS** - name server
- **A** - name to IPv4 address
- **AAAA** - name to IPv6 address
- **CNAME** - name to another name
- **PTR** - address to name
- **MX** - mail server
- and a bunch of others

Address resource records

- name ttl IN A address
 - if name ends in a period, it is used verbatim, otherwise the domain name is appended
 - ttl is optional
- Example in colorado.edu
 - simlab 3600 IN A 128.138.73.46

Start of Authority

- Typically the first record in a zone file
- Defines
 - Primary authoritative server
 - Email of DNS administrator
 - Timeouts for secondary servers
 - Default time-to-live

```
dig SOA colorado.edu
```

- `colorado.edu.` 3600 IN SOA
 `boulder.colorado.edu.` `dnsadmin.colorado.edu.`
 `2022102815` 3600 1200 2419200 3600
 -
- **SOA for `colorado.edu`**
 - TTL 1 hour
 - Serial `yyyymmddhh`
 - Refresh 1 hour
 - Retry 20 minutes
 - Expire 28 days
 - Negative cache 1 hour
- **Primary DNS server**
`boulder.colorado.edu`
- **Administrator email**
`dnsadmin@colorado.edu`

Glue Records

- Need to define name servers for domain
- Need to specify IP for name servers
- Example: (@ is shorthand for domain)

```
@ IN SOA boulder.colorado.edu. ... (...)  
      IN NS oldduke.colorado.edu.  
      IN NS boulder.colorado.edu.  
      IN NS otis.colorado.edu.
```

```
otis      IN A 128.138.129.76  
boulder   IN A 128.138.240.1  
oldduke   IN A 128.138.130.30
```

CNAME records

- Canonical name or alias
- Translates one name into another name
 - machinee IN CNAME nfs.dundermifflin.com.
- Typically requires more information to find IP
 - www.colorado.edu. IN CNAME pantheon.map.fastly.net.
pantheon.map.fastly.net. IN A 151.101.70.133
- Note that pantheon.map.fastly.net address record may be in a completely different zone file
- Pay attention to trailing periods!!!

MX records

- Mail Exchange
- Server priority
 - Lower is first
 - Try in order
- What is the IP?
- Today mail also needs an SPF record
 - Sender Policy Framework

```
dig MX gmail.com
;; QUESTION SECTION:
;gmail.com. IN MX

;; ANSWER SECTION:
gmail.com. 3600 IN MX 10 alt1.gmail-smtp-in.l.google.com.
gmail.com. 3600 IN MX 40 alt4.gmail-smtp-in.l.google.com.
gmail.com. 3600 IN MX 30 alt3.gmail-smtp-in.l.google.com.
gmail.com. 3600 IN MX 5 gmail-smtp-in.l.google.com.
gmail.com. 3600 IN MX 20 alt2.gmail-smtp-in.l.google.com.

;; AUTHORITY SECTION:
gmail.com. 172800 IN NS ns2.google.com.
gmail.com. 172800 IN NS ns1.google.com.
gmail.com. 172800 IN NS ns3.google.com.
gmail.com. 172800 IN NS ns4.google.com.

;; ADDITIONAL SECTION:
ns1.google.com. 99745 IN A 216.239.32.10
ns2.google.com. 99745 IN A 216.239.34.10
ns3.google.com. 99745 IN A 216.239.36.10
ns4.google.com. 99745 IN A 216.239.38.10

dig TXT gmail.com
gmail.com. IN TXT "v=spf1 redirect=_spf.google.com"
```

PTR records

- address IN PTR name
 - Used for reverse DNS
 - Address is in reverse octet order
- Example:
 - **otis.colorado.edu = 128.138.129.76**
76.129.138.128.in-addr.arpa. IN PTR otis.colorado.edu.
 - **Entry in zone file for domain 128.138.129.0/24**
129.138.128.in-addr.arpa. SOA (...)
76 IN PTR otis.colorado.edu.

Configuring BIND

- `options` defines global options
 - can override for individual zones and views
- `logging` configures logging
- `zone` defines a domain or subdomain
- `view` defines zones to show for split DNS
- Many more, mostly for maintainability
 - ACL, masters, etc

Key BIND options

- `listen-on` sets IP and port for server
- `recursion` enable recursion
- `allow-recursion` addresses from which to allow recursion
- `allow-query` addresses that may query
- `allow-transfer` addresses that may transfer entire domain
- `allow-update` addresses that may update zones
- `also-notify` send zone update notifications
- `match-clients` sets access to view
- MANY more

BIND zones

- Zone **type**
 - hint (e.g. root servers)
 - primary - authoritative
 - secondary - copy
 - Requires primaries
 - forward - cache only
- Zone **file**
 - Reference copy
 - Maintain this on primary
 - Store in /etc/named or /etc/bind or /etc/bind9
 - Secondary cache
 - Binary file
 - Used on restart
 - Stored on /var

Example named.conf

- **Primary**

```
options {  
    listen-on port 53 { any; };  
    directory "/var/named";  
    allow-query { any; };  
    allow-recursion { 100.64.42.0/24; };  
    recursion on;  
}  
zone "." IN {type hint; file "named.ca"; };  
zone "dundermifflin.com." IN {type primary; file "/etc/named/db.dm"; };  
zone "42.64.100.in-addr.arpa" IN {type primary; file "/etc/named/db.100.64.42"; };  
include "/etc/named.rfc1912.zones";
```

- **Secondary**

```
options {  
    listen-on port 53 { any; };  
    directory "/var/cache/bind";  
    allow-query { any; };  
    allow-recursion { 100.64.42.0/24; };  
    recursion on;  
};  
include "/etc/bind/named.conf.default-zones";  
zone "dundermifflin.com." IN {type secondary;primaries {100.64.41.2;};file "db.dm";};  
zone "42.64.100.in-addr.arpa" IN {type secondary;primaries {100.64.41.2;};file "db.100.64.42";};
```

Example Forward DNS Zone file

- Zone is dundermifflin.com.

```
$TTL 1h
@ IN SOA ns1.dundermifflin.com. admin.dundermifflin.com. (
    20221017 ; serial
    3h ; refresh
    1h ; retry
    14d ; expire
    1h); negative cache
IN NS ns1.dundermifflin.com.
IN NS ns2.dundermifflin.com.
IN MX 10 mail.dundermifflin.com.

ns1      IN A 100.64.42.2
ns2      IN A 100.64.42.6
router   IN A 100.64.0.42
dmz     IN A 100.64.42.1
www 10m IN CNAME web1.dundermifflin.com.
```

Example reverse DNS file

- Zone is 42.64.100.in-addr.arpa (domain is 100.64.42.0/24)

```
$TTL 1h
@ IN SOA ns1.dundermifflin.com. admin.dundermifflin.com. (
    20221017 ; serial
    3d        ; refresh
    1h        ; retry
    14d       ; expire
    1h)       ; negative cache
IN NS ns1.dundermifflin.com.
IN NS ns2.dundermifflin.com.

1 2h IN PTR dmz.dundermifflin.com.
2 2h IN PTR ns1.dundermifflin.com.
3 2h IN PTR web0.dundermifflin.com.
6 2h IN PTR web0.dundermifflin.com.
```

Reducing BIND chatter

- When `named` starts there is a LOT of stuff in the logs which can make errors hard to find
- Extraneous messages is reduced by
 - running `named` with the `-4` option (IPv4 only)
 - `empty-zones-enable no`
- When errors occur `grep named` from `/var/log/messages` or `/var/log/syslog` so you can see **all** the output
 - With `named` errors are rarely the last thing in the log

systemd-resolved

- Another of systemd's attempts at taking over the world
 - Solution in search of a problem
- Replaces `/etc/resolv.conf` with a symlink
- Maps to 127.0.0.53
 - `dig` shows 127.0.0.53 as DNS server
- `resolvectl` shows DNS servers and search

Network File System

Linux System Administration
Fall 2023

NFS

- Disk access across the network
- Designed by Sun in 1984
 - Storage for diskless workstations
- De facto standard for Unix/Linux
 - NFSv2 1989 - based on UDP
 - NFSv3 1995 - use TCP for congestion
 - NFSv4 2003-2015 - performance and security

NFS Alternatives

- SMB/CIFS (using Samba on Linux)
 - Windows networking file system
- GlusterFS
 - High performance cluster distributed filesystem
 - Load balancing, failover, snapshots, etc.
- Many others

NFS Server

- **Install and start**
 - `dnf install nfs-utils`
- **Set directories to export in /etc/exports**
 - `/dir subnet (options)`
- **Enable and Start server**
 - `systemctl enable nfs-server`
 - `systemctl restart nfs-server`

Managing exports

- Export /var/ftp to 100.64.42.0/24
 /var/ftp 100.64.42.0/24 (rw, sync, no_root_squash)
- Can list several subnet (option) space separated
- Set options by subnet
- Refresh with command exportfs -r
- Can also add exports to /etc/exports.d
- NFS4 should export trees, but mostly does not

export options

- ro **read only**
- rw **read/write**
- root_squash **UID0=anon**
- no_root_squash
- all_squash **UID=anon**
- no_all_squash
- anonuid=
- anongid=
- sync **wait for disk**
- async
- secure_locks
- insecure_locs

Testing nfs

- `nfsstat` Shows server and client usage
 - Call stats and types
- `showmount -e` Shows directories exported
 - Run on B as `showmount -e 10.21.32.2`
- `mount` Just try it
 - `root_squash` will bite you

NFS user mapping

- UID or username?
GID or group?
 - NFS 2&3 used only UID & GID
 - NFS4 uses identity mapping (username and group)
- Identity mapping used in translating uid/gid to names but not for authentication (file access)
 - The **server** controls permissions
- Sanity requires consistent username/UID and group/ GID across systems
- Linux defaults to domain name (*so you can ignore this in Lab 11*)

NFS client

- Install with `apt install nfs-common`
- Mount with `mount host:/dir /localdir`
- Mounts are just like a physical disk or LV, but mount options are somewhat different
- **If the server dies, the client will hang**
 - Use `autofs` to mount as needed

nfs mount options

- `rw` read/write
- `ro` read only
- `bg` try in background
- `hard` block on fail
- `soft` error on fail
- `intr` interruptable
- `nointr` not interruptable
- `retrans` tries before fail
- `timeo` timeout
- `rsize` read buffer size
- `wsize` write buffer size
- `sec` security flavor
- `proto` UDP or TCP
- `nfsvers` NFS version

automount

- Mount the directory tree when you access it
 - Slight delay on first access, unmount on idle
 - Minimizes client hang when server dies
- `autofs` is the kernel implementation
 - `/etc/auto.master` is the master map
 - `/etc/auto.*` are other maps

automount maps

- master defines direct and indirect maps
- direct map mounts ad hoc directories
 - Example: /var/mail
- indirect maps mount to a shared directory
 - Example: /raid/* and /net -hosts
 - NIS maps

Lab 11 hints

- KISS - nothing fancy is required
- Machine E
 - Create directory to export
 - Install NFS, set the export, enable the server
- Machine C (and D)
 - Create mount point (only on C)
 - Figure out access
 - Install NFS, mount the directory, make mount permanent
 - Extra credit: set up autofs

Network Time Protocol

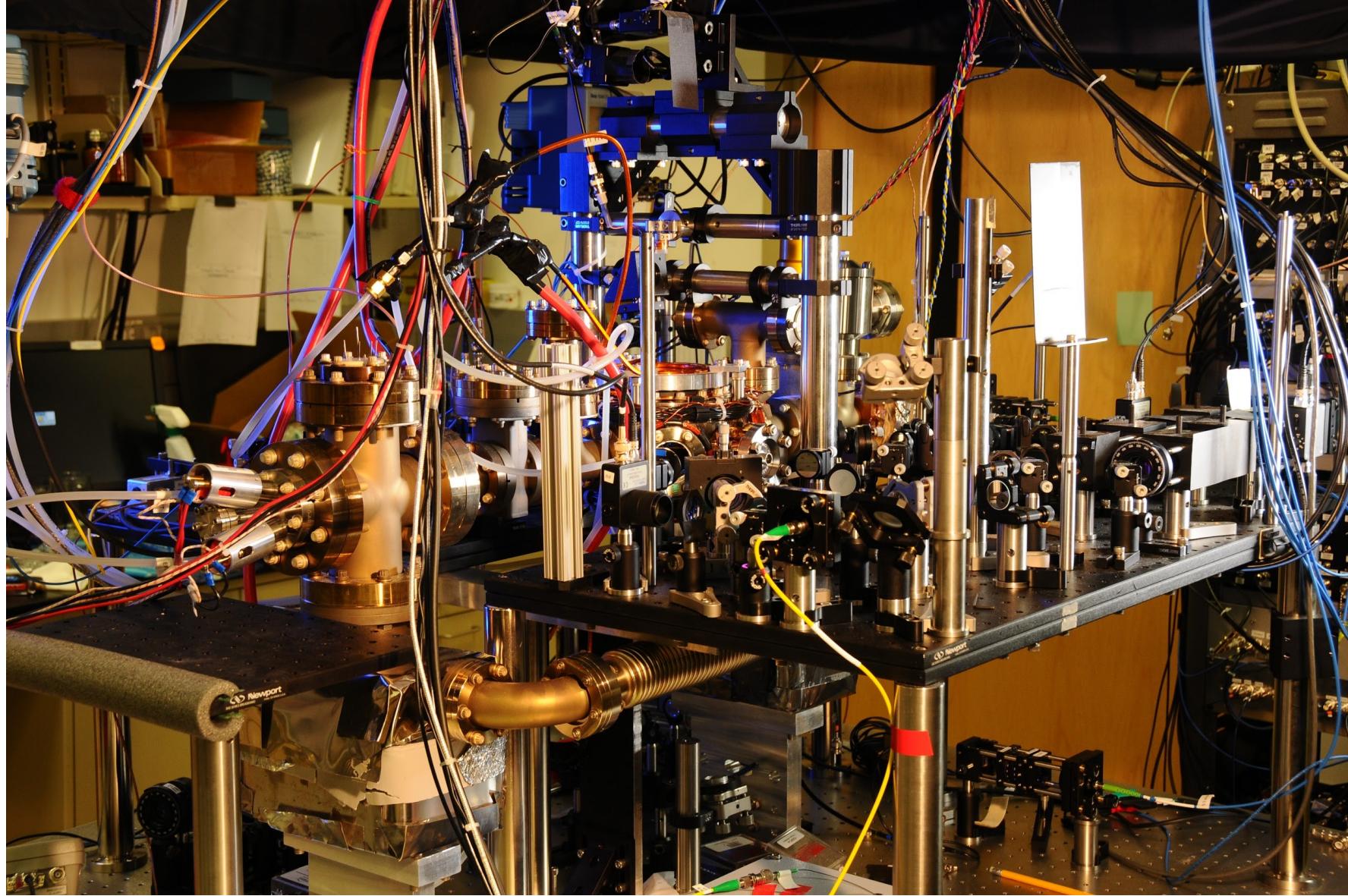
Linux System Administration
Fall 2023

The need for NTP

- Systems need to agree on the time
 - Time stamps on files
 - Critical for networked file systems
 - Time stamps in logs
 - Critical for debugging distributed problems
 - Time stamps on transactions
 - Critical for trading, banking, etc
- Common time is good, absolute time is best

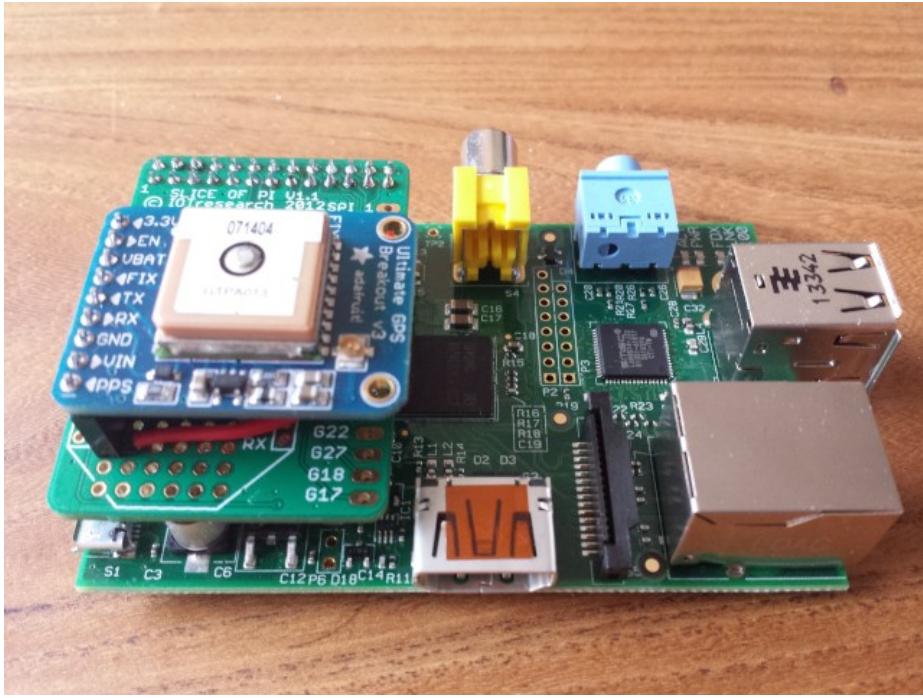
Time Strata

- 0: atomic clocks (also GPS/GNSS)
- 1: Computers slaved to Startum 0 clocks
- 2: Computers network synced to Stratum 1
- 3-15: Computers network synced to higher strata



Time Sources (PPS or Frequency)

- sub microsecond
 - Caesium clocks
 - GPS (USA)
 - Galileo (Europe)
 - GLONASS (Russia)
 - BeiDou (China)
- sub millisecond
 - Rubidium clock
 - WWV (USA)
 - DCF (Europe)
 - JJY (Japan)
 - Cellular



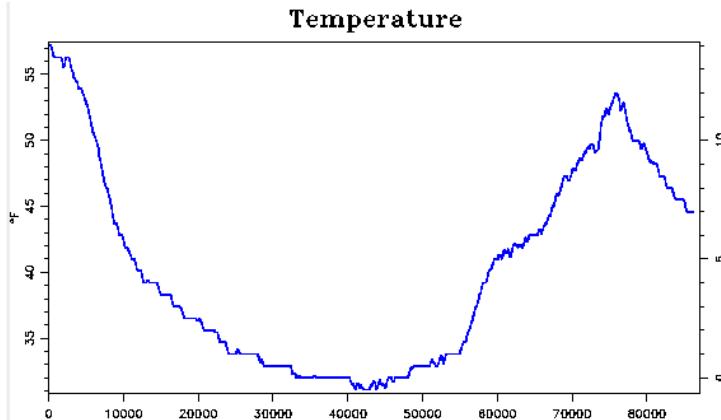
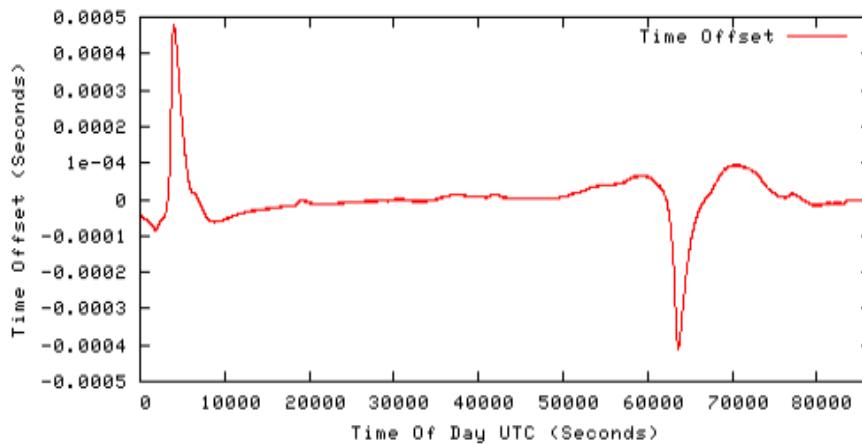
Temperature Sensitivity

NTP Status:

Status	In Sync
Stratum	1
Selected Reference	.GPS.
Delay	0.000 mS
Offset	0.048 mS
Jitter	0.002 mS

Offset Graph:

04/11/2023 Time Offset
 Frequency Offset
 RMS Jitter
[Refresh Graph](#)



NTP Reference Status:

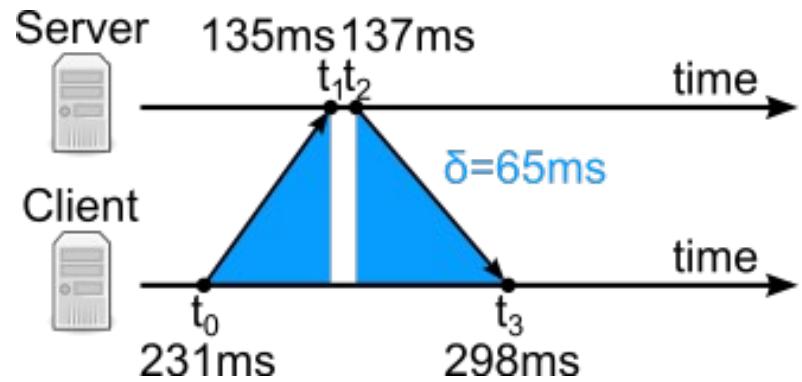
Sync	Host	Ref ID	Stratum	Mode	Type	Auth Status	Last (Sec)	Poll Interval (Sec)	Reach	Delay (mS)	Offset (mS)	Jitter (mS)
*	Spectracom	.GPS.	0	Client	local	none	10	16	377	0.000	0.117	0.002

WWV & WWVB Fort Collins CO



NTP Protocol

- 64 bit time stamp (32 bit seconds, 32 bit fraction)
- Estimate network delays
 - Asymmetric network delay
 - $dt = [(t_1 - t_0) + (t_2 - t_3)] / 2$
- Adjust clock frequency to sync
 - Gradually get clocks in sync
 - Estimate clock drift



Initial clock synchronization

- Hardware clock
- Network request (TCP or UDP)
 - daytime (text string) TCP or UDP port 13
 - time (32 bit UNIX epoch) TCP or UDP port 37
- NTP based
 - ntpdate
 - sntp

chrony (RedHat NTP)

```
chronyc -N 'sources -a -v'
```

/etc/chrony.conf

- Get time from public NTP pool with initial burst
 - pool 2.pool.ntp.org iburst
- Get time from specific servers
 - server time-a-www.nist.gov
 - server time-a-b.nist.gov
- Allow requests from subnets
 - allow 100.64.42.0/24

systemd-timesyncd

- Default on debian
- Configure with /etc/systemd/timesyncd.conf
- Will use values from DHCP
- View status with systemctl

```
systemctl status systemd-timesyncd
systemd-timesyncd.service - Network Time Synchronization
  Loaded: loaded (/lib/systemd/system/systemd-timesyncd.service);
            enabled; vendor preset: enabled
  Active: active (running) since Sat 2022-10-29 16:55:25 MDT; 2 days ago
    Docs: man:systemd-timesyncd.service(8)
   Main PID: 436 (systemd-timesyn)
     Status: "Initial synchronization to time server 132.163.96.1:123
              (132.163.96.1)."
        Tasks: 2 (limit: 1132)
```

Other NTP servers

- ntpd - the original
 - xntpd - NTPv3
- OpenNTP - OpenBSD implementation
- PTP - Precision Time Protocol
 - sub microsecond precision

Leap Seconds

- 61 or 59 second minutes on 31-Dec 23:59 UTC
 - Skip 23:59:59 (miss a second)
 - Add 23:59:60 (add a second)
- Confuses many systems(including UNIX epoch)
 - step clock to add offset
 - slew or smear clock

Routing

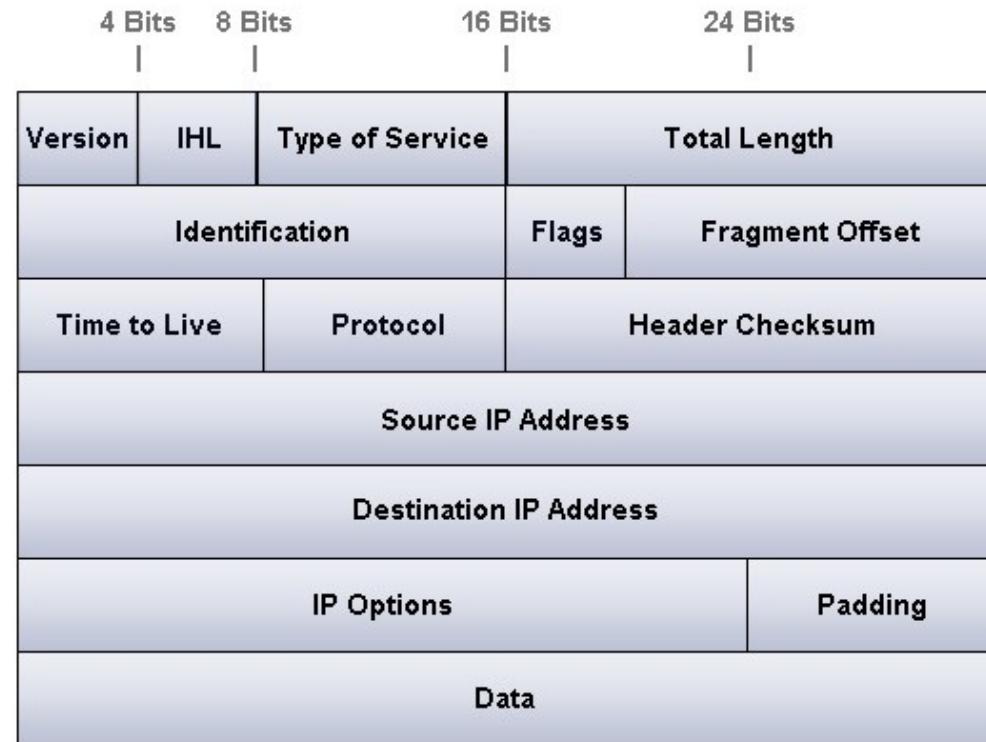
Linux System Administration
Fall 2023

Debugging Routing

- ping
 - Tests end-to-end connectivity
- traceroute **or** (or tracert on Windoze)
 - Shows the route the packet takes
 - Linux traceroute use UDP packets
- ip route **or** route **or** netstat -nr
 - Configure or show routing

Anatomy of an IP packet

- Always big-endian
- Version=4
- IHL=IP Hdr Len
- Type of Service
 - Min delay
 - Max throughput
- Flags & Frag Off
 - Large packets
- Protocols add additional header in data section



Configuring an IPv4 interface

- Must have
 - IP address
 - Netmask
 - Default gateway
- Nice to have
 - DNS server address
 - NTP server address
- Example:
 - IP 100.64.0.42/24
 - GW 100.64.0.254
 - DNS1 128.138.240.1
 - DNS2 128.138.130.30
 - NTP 132.163.97.1
 - time-a.www.nist.gov

Routing

- Router connects multiple subnets
- The next hop must be reachable at Layer 2
- Each subnet has an associated target
 - via (IP address of gateway)
 - Packet sent to the MAC address of the gateway
 - interface
 - Packet sent to the MAC address of destination
- Selects **best** match out of all possibilities

Routing Example

- Destination IP in packet 100.64.42.6
- Routing Table (*gateway and MAC not shown*)

ifce	subnet	remarks
eno1	0.0.0.0/0	match 0 bits (default gateway)
eno2	128.0.0.0/8	does not match 1 st octet
eno3	100.0.0.0/8	match 8 bits
eno4	100.64.0.0/16	match 16 bits
eno5	100.64.42.0/24	match 24 bits (best match)
eno6	100.64.10.0/24	does not match 3 rd octet
eno7	100.64.42.8/29	does not match 5 bits in 4 th octet

- Packet sent over interface eno5

Example: Machine A

- Interfaces

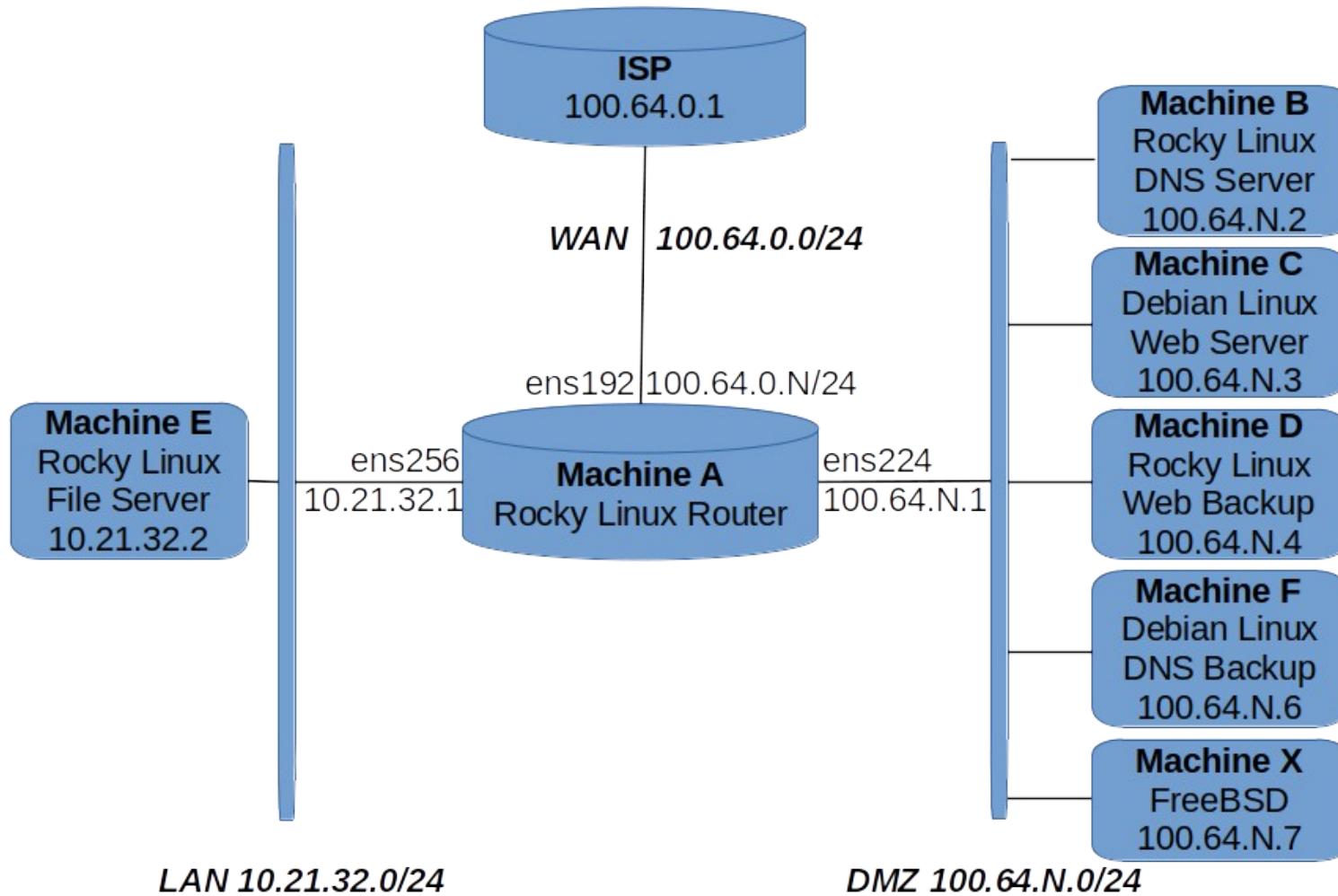
```
ens192 WAN IP 100.64.0.42/24 NET 100.64.0.0/24      gateway 100.64.0.254
ens224 DMZ IP 100.64.42.1/24 NET 100.64.42.0/24
ens256 LAN IP 10.21.32.1/24  NET 10.21.32.0/24
```

- No gateway for DMZ and LAN because Machine A is actually the gateway
- All routes except default added by kernel

```
ip route
default via 100.64.0.254 dev ens192 proto static metric 100
10.21.32.0/24 dev ens256 proto kernel scope link src 10.21.32.1 metric 102
100.64.0.0/24 dev ens192 proto kernel scope link src 100.64.0.42 metric 100
100.64.42.0/24 dev ens224 proto kernel scope link src 100.64.42.1 metric 101
```

- NAT LAN to WAN (/etc/sysconfig/nftables.conf)

```
table ip nat {
    chain POSTROUTING {
        type nat hook postrouting priority srcnat; policy accept;
        oifname "ens192" ip saddr 10.21.32.0/24 masquerade
    }
}
```



traceroute

- Run on N=41 Machine C 100.64.41.3
- traceroute 100.64.42.3

1	100.64.41.1	N=41 Machine A DMZ
2	100.64.0.254	ISP Gateway
3	100.64.0.42	N=42 Machine A WAN
4	100.64.42.3	N=42 Machine C

- traceroute 100.64.0.42

1	100.64.41.1	N=41 Machine A DMZ
2	100.64.0.42	N=42 Machine A WAN

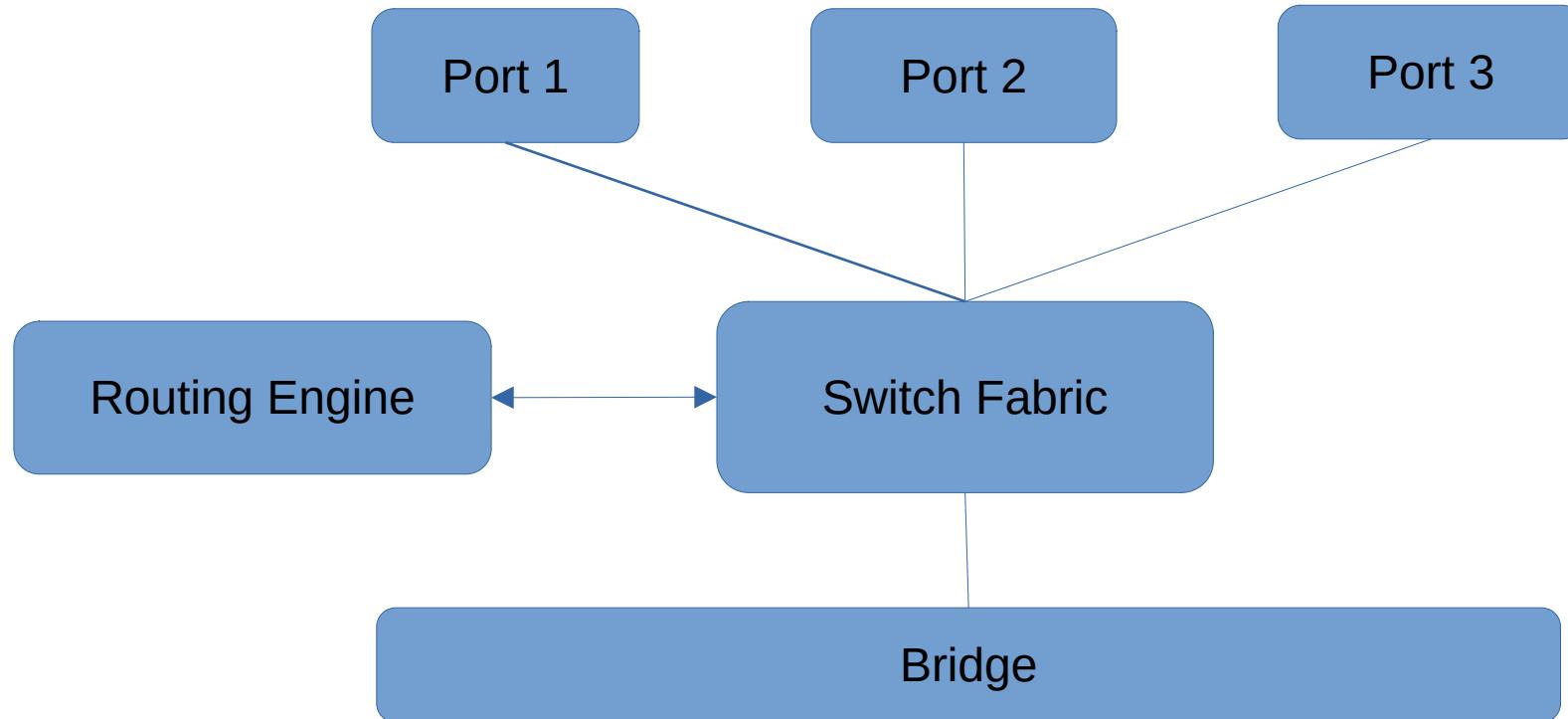
Dynamic Routing

- OSPF **Open Shortest Path First**
 - Interior gateway protocol
 - Weighted link state Routing
 - Mostly used in enterprise level interior networks
- BGP **Border Gateway Protocol**
 - Exterior gateway protocol
 - Path-vector routing
 - Runs most of the internet

Linux routing

- Static routing using the kernel
 - /etc/sysctl.conf
net.ipv4.ip_forward = 1
- Dynamic routing with Quagga (fork of GNU Zebra)
 - ospfd, ospf6d - OSPF 2 & 3
 - bgpd - BGP
- Without a forwarding plane, throughput is limited by CPU
 - Use a hardware router instead

Anatomy of a router



Network Address Translation (NAT)

- Port Address Translation (masquerade)
 - Remaps address and port
10.21.32.2:9876 becomes 100.64.42.1:7654
 - Outbound:
 - Replace source IP and port with public IP and port
 - Inbound to NAT port:
 - Replace destination IP and port with private IP and port
- ICMP or port-less protocols use other packet data

Firewalls

Linux System Administration
Fall 2023

The purpose of Firewalls

- Control packet processing
 - Input - packets destined for a local address
 - Output - packets with source a local address
 - Forward - packets with source and destination that are not local
- Enforce connectivity policy

Defense in Depth

- Filter at the router
 - Limit access to DMZ and LAN
- Filter at each machine
 - Limit DMZ machine functionality
 - Trust devices on the LAN

Security is hard

- Don't write firewall rules if you don't know what you are doing
 - Don't assume that after this class, you can write firewall rules!!!
- Be paranoid
 - Don't trust the network

Linux Firewalls

- ipchains
 - The original 1990s (stateless)
- iptables
 - Mainstay of most firewalls (1998-2022)
- nftables
 - new hotness (2014-)
- netfilter
 - Kernel module that implements packet filtering

Wrappers around iptables/nftables

- firewalld Redhat
- ufw uncomplicated firewall Debian
- shorewall simplified rule generator
- fail2ban dynamic blocking
- pfSense popular BSD firewall

Testing firewalls

- ping & traceroute basic connectivity
- nmap network scanner
 - scan for TCP on 100.64.42.0/24

```
nmap -sS 100.64.42.0/24
```
 - fast scan all ports on 100.64.42.3

```
nmap -T4 -n -Pn 100.64.42.3
```
- ***It matters where you run the test from***

nmap -n -sS -sU 100.64.42.0/24

Nmap scan report for 100.64.42.1

Host is up (0.040s latency).
Not shown: 1997 closed ports
PORT STATE SERVICE
22/tcp open ssh
67/udp open|filtered dhcps
123/udp open|filtered ntp

Nmap scan report for 100.64.42.2

Host is up (0.037s latency).
Not shown: 1997 closed ports
PORT STATE SERVICE
22/tcp open ssh
111/tcp open rpcbind
111/udp open rpcbind

Nmap scan report for 100.64.42.3

Host is up (0.040s latency).
Not shown: 1995 closed ports
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
111/tcp open rpcbind
68/udp open|filtered dhcpc
111/udp open rpcbind

Nmap scan report for 100.64.42.4

Host is up (0.034s latency).
Not shown: 1996 closed ports
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
111/tcp open rpcbind
111/udp open rpcbind

Nmap scan report for 100.64.42.6

Host is up (0.037s latency).
Not shown: 1994 closed ports
PORT STATE SERVICE
22/tcp open ssh
53/tcp open domain
111/tcp open rpcbind
53/udp open domain
68/udp open|filtered dhcpc
111/udp open rpcbind

Stateless vs. Statefull

- Stateless
 - Packets define what is allowed
 - Simplifies work for developers and improves performance
- Statefull (connection tracking)
 - Decision may depend on previous packets
 - Greatly simplifies writing firewall for user
 - Allow out what was allowed in
 - Absolutely required for complex protocols like ftp

nftables vs. iptables

- Same team of developers
 - Same kernel module (netfilter)
 - Same terminology
- New syntax in nftables
 - More flexible rules
- No default chains in nftables
 - Improves performance and scalability
- Comprehensive types in nftables
 - Includes ARP, IPv6, ethernet

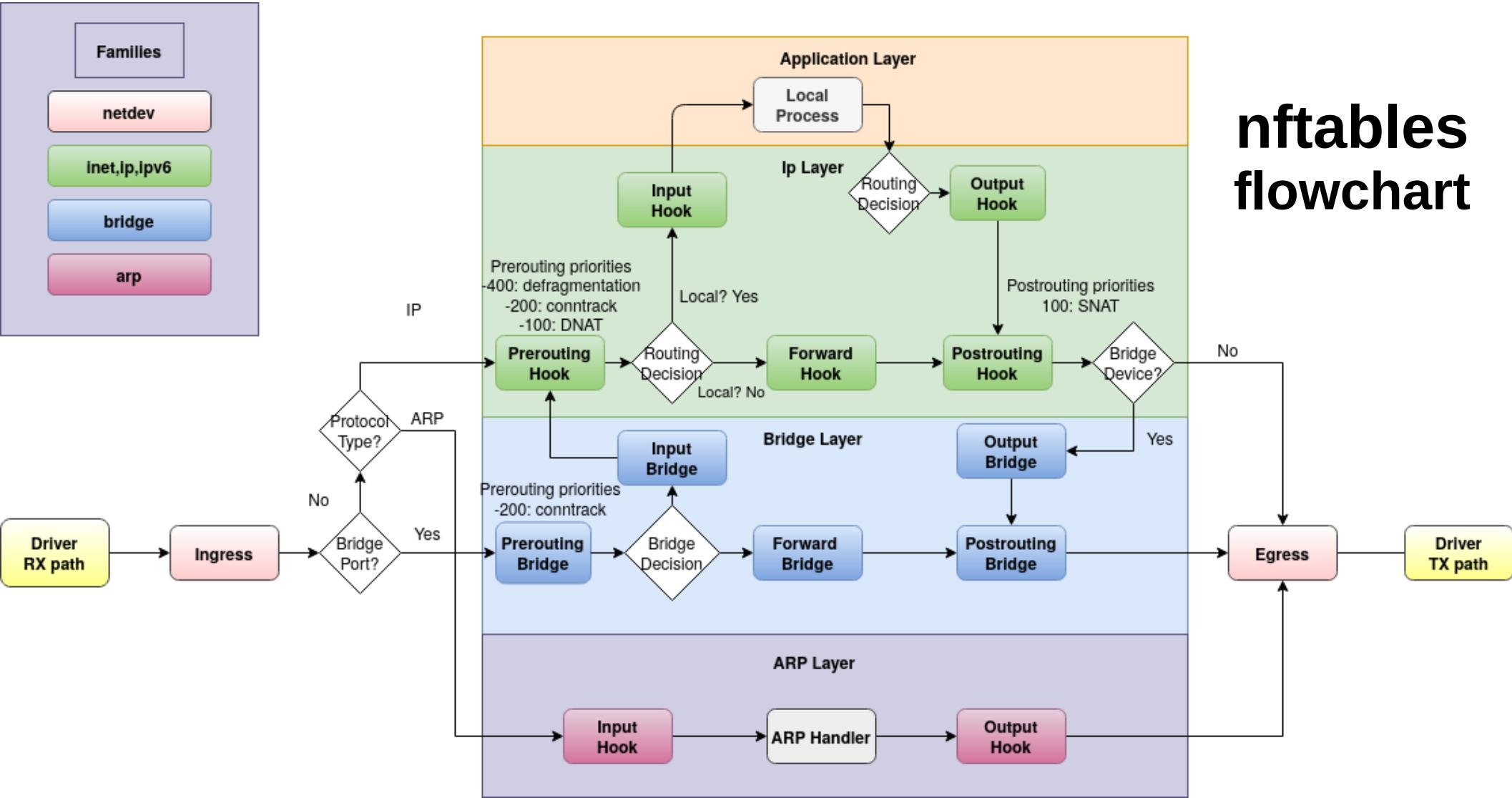
iptables terminology

- table **container for chains**
 - Predefined names filter, nat, mangle, raw
- chain **list of rules executed in sequence**
 - filter table has INPUT, OUTPUT and FORWARD
 - nat table has PREROUTING, POSTROUTING and OUTPUT
 - Default action is called policy
- rule **defined action**
 - Actions are ACCEPT, DROP, QUEUE or RETURN

nftables terminology

- table container for chains
- chain list of rules
 - type filter, route or nat
 - hook where this chain fits
 - priority used to order chains
 - policy default action
- rule defined action

nftables flowchart



Example: local access to ssh server

- Accept only TCP/IP connections on port 22 from LAN and DMZ
- Allow any outgoing connections
- iptables

```
- iptables -A INPUT -p tcp --dport 22 -s 100.64.42.0/24 -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -s 10.21.32.0/24 -j ACCEPT
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
```

- nftables

```
- nft -i
table ip saclass {
    chain incoming {
        type filter hook input priority 0; policy drop;
        ip saddr {100.64.42.0/24,10.21.32.0/24} tcp dport 22 accept
    }
}
# No output chain - defaults to accept
```

Understanding iptables/nftables

- Packets mapped to table & chains based on source
 - filter input, output, forward, etc
- Rules in chain are executed in order
 - check for match of address, port, etc
 - action taken if match
 - rest of rules are not evaluated
 - except if the action was jump
- You have to allow replies to establish a connection

tables

- The table select the packet family
- iptables predefines filter, nat, mangle, raw
- nftables user defined table *family* name
 - ip IPv4
 - ip6 IPv6
 - inet Internet (IPv4 or IPv6)
 - arp IPv4 ARP
 - bridge bridge (ethernet)
 - Example: nft add table ip saclass

base chains

- iptables **predefined** filter chains INPUT, OUTPUT and FORWARD
 - Policy set by iptables -P chain action
Example: iptables -P INPUT DROP
- nftables **user defined** table chain name
 - type **chain type** filter, nat, route
 - hook **where it fits** input, forward, output, prerouting, postrouting
 - priority **order of chains (low to high)** numeric or by name
 - policy **default action** accept or drop
 - Example: Incoming packet filter drop except what is allowed

```
nft add chain saclass incoming
'{type filter hook input priority 0; policy drop;}'
```

rules

- Rules are executed in order
- Execution stops when a rule matches and action is taken
 - iptables actions are ACCEPT, DROP, REJECT
 - nftables actions are accept, drop, reject
- Example: Allow incoming to port 22 (ssh)
 - iptables

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```
 - nftables

```
nft add rule saclass incoming tcp dport 22 accept
```
- All the conditions in the rule must match

iptables

- `-F` flush (clear)
- `-A` append rule
- `-I` insert rule
- `-D` delete rule
- `-N` new chain
- `-P` set policy
- Set rules on boot
 - Redhat
 - `/etc/sysconfig/iptables`
 - Debian
 - `/etc/iptables/rules.v4`
- Show rules
 - `iptables -nvL`

Example /etc/sysconfig/iptables

- #iptables for dundermifflin web server
 - *filter
 - # Default drop INPUT, accept FORWARD and OUTPUT
 - :INPUT DROP [0:0]
 - :FORWARD ACCEPT [0:0]
 - :OUTPUT ACCEPT [0:0]
 - # Allow established or related connections
 - A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
 - # Allow selected icmp traffic
 - A INPUT -p icmp --icmp-type echo-request -j ACCEPT
 - A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
 - A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT
 - A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
 - A INPUT -p icmp -j DROP
 - # Allow any inbound http
 - A INPUT -p tcp --dport 80 -j ACCEPT
 - # Permit selected inbound ssh
 - A INPUT -p tcp -s 198.18.0.0/16 --dport 22 -j ACCEPT
 - A INPUT -p tcp -s 100.64.0.0/16 --dport 22 -j ACCEPT
 - A INPUT -p tcp -s 10.21.32.0/24 --dport 22 -j ACCEPT
 - # Deny outbound to facebook
 - A OUTPUT -d 157.240.28.35 -j DROP
- COMMIT

nft (nftables)

- list **show content**
 - ruleset
 - table
 - chain
- add **item**
- delete **item**
- flush **empty**
- Set rules on boot
 - Redhat
 - /etc/sysconfig/nftables.conf
 - Debian
 - /etc/nftables.conf
- Show all rules
 - nft list ruleset

Example /etc/sysconfig/nftables.conf

```
•#!/usr/sbin/nft -f

flush ruleset

table ip saclass {
    chain incoming {
        type filter hook input priority 0; policy drop;
        # Allow established or related connections
        ct state related,established accept
        # Allow selected icmp traffic
        icmp type {echo-reply,destination-unreachable,echo-request,time-exceeded} accept
        # Allow any inbound http
        tcp dport 80 accept
        # Permit selected inbound ssh
        ip saddr {10.21.32.2/24,198.18.0.0/16} tcp dport 22 accept
    }
    chain outgoing {
        type filter hook output priority 0; policy drop;
        # Block facebook
        ip daddr 157.240.28.35 drop
    }
}
# No forward chain which allows everything
```

Connection Tracking ct

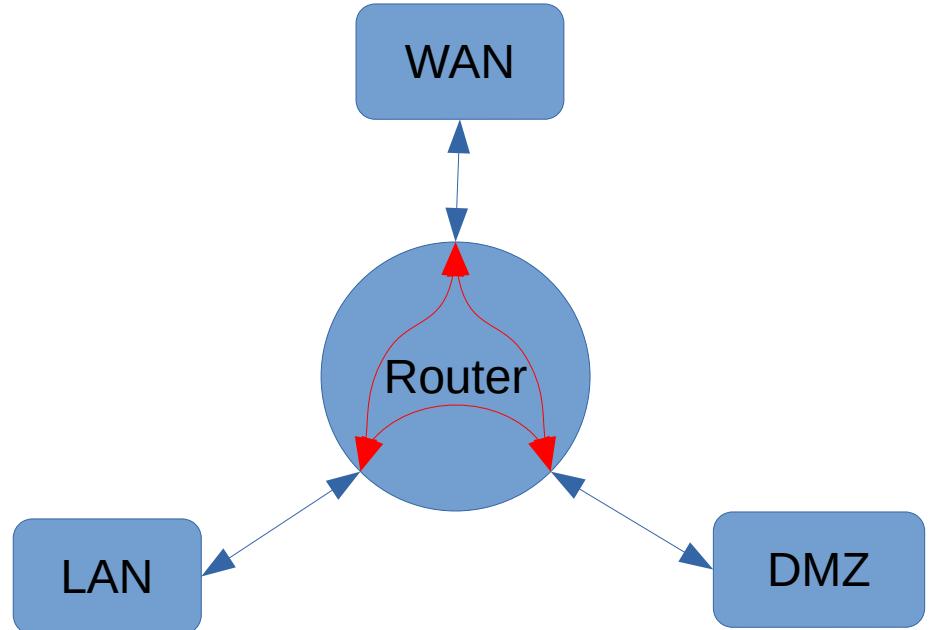
- Statefull firewall
- Allows replies to permitted connections
- Critical for complex protocols like ftp
- Typical usage
 - ct state established, related accept
 - ct state invalid drop

Machine A Firewall

- Inbound connections
 - ssh from LAN, DMZ, WAN and VPN
 - DHCP and NTP from LAN and DMZ
 - returns from outbound connections
- Outbound connections
 - Any except facebook

Machine A Forwarding

- Three interfaces
 - **LAN, DMZ, WAN**
- Six paths
 - **WAN to LAN**
 - **WAN to DMZ**
 - **DMZ to WAN**
 - **DMZ to LAN**
 - **LAN to WAN**
 - **LAN to DMZ**

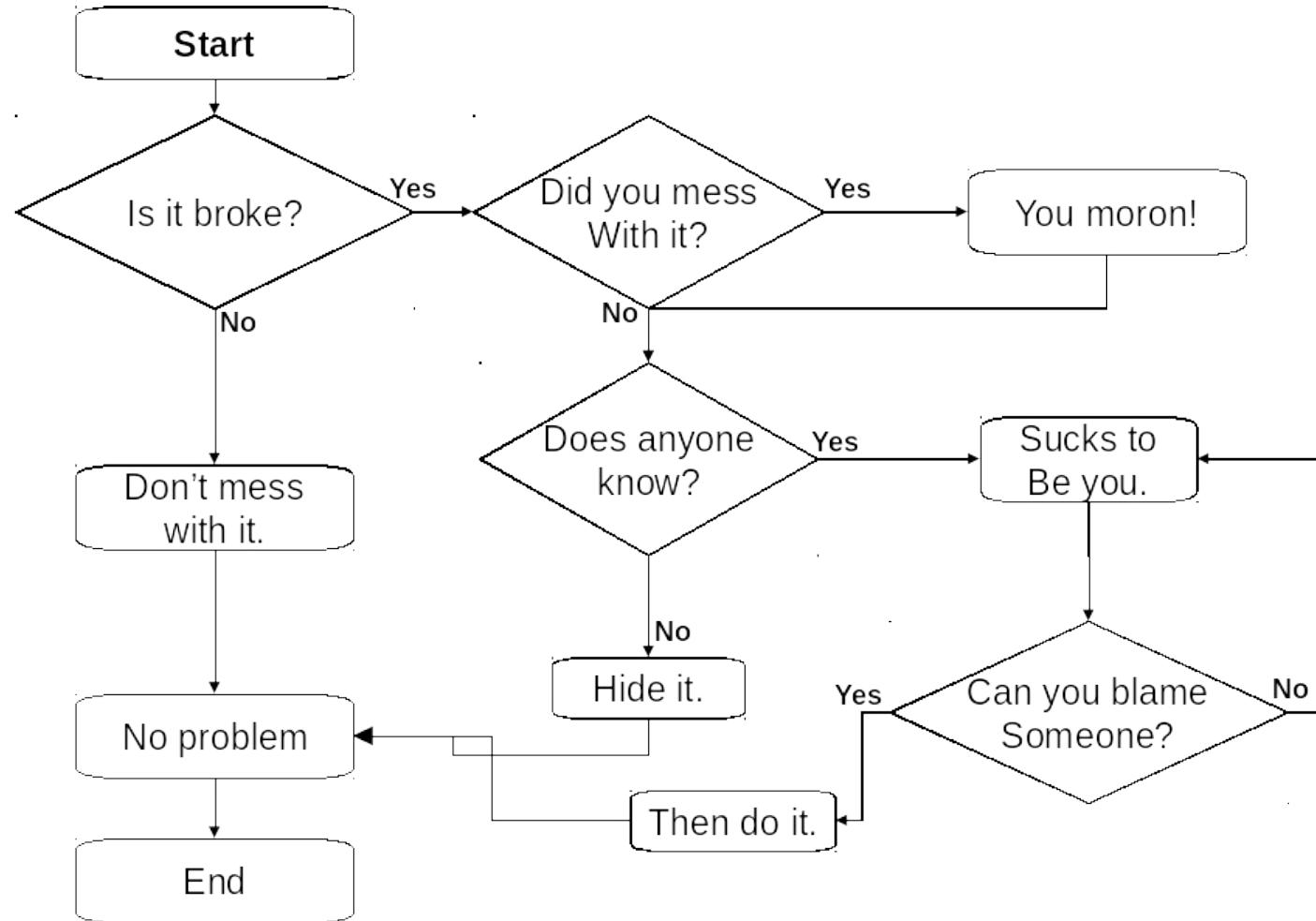


Machine A Forward Chains

- **WAN to DMZ**
 - ICMP
 - UDP DNS
 - TCP ssh, http, grader
 - return packets
- **WAN to LAN**
 - return packets
- **LAN to WAN**
 - anything except facebook
- **LAN to DMZ**
 - anything
- **DMZ to WAN**
 - ICMP
 - UDP DNS
 - TCP DNS, http, https
 - return packets
- **DMZ to LAN**
 - ICMP
 - TCP ssh,NFS
 - return packets

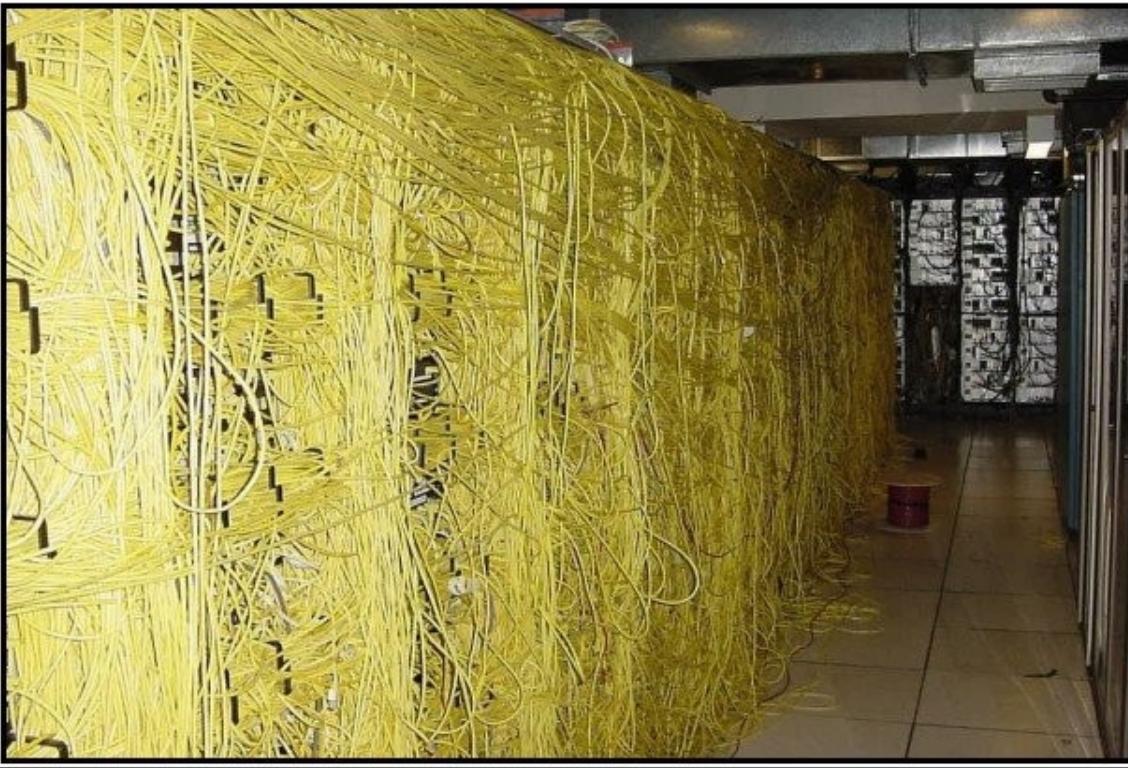
Troubleshooting

Linux System Administration
Fall 2023



Basic Troubleshooting

- Does it have power?
- Is it powered on?
- Is there system activity?
- Is there console activity?
- Is it plugged into the network?
- Did you recently make changes?



A NETWORK CABLE IS UNPLUGGED

Telecom Lab Final

Troubleshooting Approach

- Think before you do
 - Don't make the problem worse
- Be methodical
 - Identify symptoms
 - Brainstorm causes
 - Formulate hypothesis
 - Test hypothesis

Avoid Cargo Cult syndrome

- The solution makes no sense but it worked last time
 - Power cycle or reboot solves any problem
 - Treating the symptom instead of finding the source of the problem



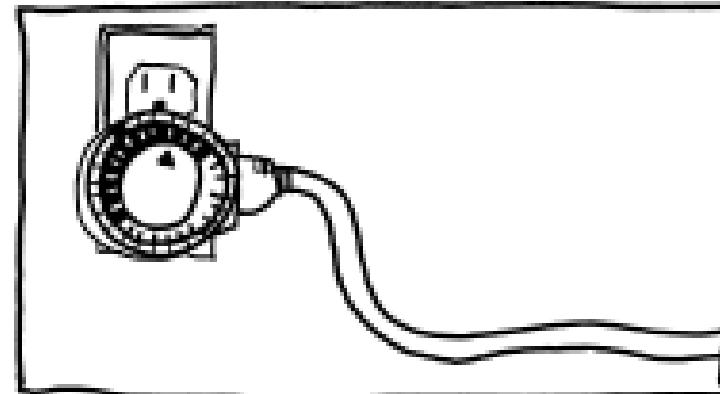
Invest time in locating the cause

FIGURING OUT WHY MY HOME
SERVER KEEPS RUNNING OUT
OF SWAP SPACE AND CRASHING:



1-10 HOURS

PLUGGING IT INTO A LIGHT TIMER
SO IT REBOOTS EVERY 24 HOURS:



5 MINUTES

WHY EVERYTHING I HAVE IS BROKEN

Memorable Errors/Solutions

- 1201 and 1202 program alarm
- SCE to AUX
- Main B Bus Undervolt
- Abort, Retry, Ignore.
- PC LOAD LETTER

Retrace your steps

- What system files did you touch recently?
 - Did you break a dependency?
- Problems with booting
 - /etc/fstab **will** hang when mounting disks
 - Default target
- Problems with networking
 - interface configuration (DHCP?)
 - routing
 - firewall

Trouble shooting power

- Is the UPS screaming?
- Is the power cord secure (both ends)?
- Are there lights on the server?
 - Which ones?
 - How about on the back?
 - What color are they?
 - Are you looking at the right server?

Trouble shooting booting

- Can you access IPMI?
- Do you have console access?
 - Is it stuck in BIOS/EFI?
 - Is it in rescue mode?
 - Is it running fsck?
 - Is it hung on mount? (bad /etc/fstab?)
 - Do you have a login prompt?

Trouble shooting networking

- Is the server up?
- Is it routing? Can you ping it?
 - Traceroute the path
- Can you ssh into the server?
- Can you access the service locally?
- Is it firewalled off?

Trouble shooting services

- Is it enabled on boot?
- Is it running?
- Can you restart it?
- Is there an error shown in the logs?
 - Bad/missing configuration file?
 - Is there a failed dependency?

Intermittent problems

- Brainstorm causes
 - Sometimes it helps to think of what must be correct for things to work vs. what could be broken.
- Look for a pattern
- Can you intentionally trigger the problem?
 - Can you rule out any causes?
- Test, measure, log, document

Trouble shooting pitfalls

- Always blaming one/the last issue
- Treating symptoms, not solving problems
- Changing more than one thing at a time
- Troubleshooting without a hypothesis
- Never figuring out the root cause

Troubleshooting Techniques

- Explain it to a 5 year old/rubber duck/grandma
 - Write an email explaining the problem
- Imagine your helper keeps asking why?
- What often goes wrong with this subsystem?
- How is this supposed to work?
- How is the current behavior different from what is supposed to be happening?

Document the Solution

- Sometimes lightning strikes in the same places



Sometimes the obvious works

- If it is smoking or on fire or water is pouring in,
pull the power
- The last thing you fooled with is the prime suspect
- After a power outage, a systematic reboot may work
 - You should fix dependencies to avoid this
 - Restarting key services may be enough

Debugging the DM network

- Recover Machine A first
- Make sure Machine A routes packets
- Verify critical services next
 - DHCP, ssh, DNS, NFS
- Verify other services
 - web, backup servers

Hints

- Be methodical
- Make sure you can recover a lost password
- Make sure you can access the console
 - VMRC works better than HTML
- Your lab notes and class slides is a handy guide
- D.F.I.U.